ESC-EN-HA-TR-2012-114

Technical Report
1169

Numerical Estimation of Information Theoretic Measures for Large Data Sets

M.B. Hurley
E.K. Kao

30 January 2013

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

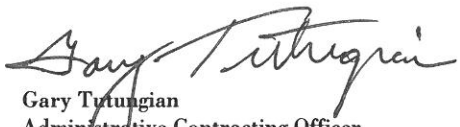
Prepared for the Department of the Air Force under Air Force Contract FA8721-05-C-0002.

This report is based on studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This work was sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002.

The 66th Air Base Group Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Gary Tutungian
Administrative Contracting Officer
Enterprise Acquisition Division

Massachusetts Institute of Technology
Lincoln Laboratory

Numerical Estimation of Information Theoretic Measures for Large Data Sets

M.B. Hurley
E.K. Kao
Group 104

Technical Report 1169

30 January 2013

Lexington Massachusetts

This page intentionally left blank.

ABSTRACT

A problem that has plagued the tracking community for decades has been the lack of a single metric to assess the overall performance of tracking systems. The community has developed many different metrics and searched for methods to fuse the multiple measures into an overall score, with each selection of metrics and fusion method usually motivated by weakly supported arguments. Without an overall metric, it was difficult to determine if conflicting differences in performance from different metrics, such as track length, track swaps, and track breaks, were leading to overall better or worse performance. For the intelligence, surveillance, and reconnaissance (ISR) community, the inability to recognize superior performance from trackers meant that it was difficult to acquire the right trackers for specific applications and to prove that performance was improving as trackers were tuned or new trackers acquired.

The authors' prior research has identified total conditional entropy as a useful measure to assess the overall performance of multi-target trackers and classifiers. The measure can be used to evaluate any system that can be formulated as an assignment algorithm that maps N classes of objects to M labels. The assignments from the decision system are compared to a truth data set to generate the total conditional entropy. Two-dimensional plots of (1) mutual information normalized by the truth entropy and (2) the conditional entropy of the sensor data given the truth data normalized by the truth entropy can be used to construct plots similar to the receiver operational characteristic (ROC) plots of detection theory. These plots are useful for graphically depicting the performance differences between tracking algorithms.

The authors' prior research neglected to provide error bounds on the information theoretic measures, which are necessary to estimate the statistical significance of results from tests on competing trackers and classifiers. This report focuses on work conducted to generate error bounds for this purpose. While reviewing literature to identify earlier work on error estimation for information theoretic measures, the papers of Wolpert and Wolf were found to provide exact equations for calculating the first- and second-order statistics for the three fundamental entropy measures from sample data. The associations in the sample data are counted to get the number of times that objects in class N are assigned labels in class M. The authors' direct conversion of the Wolpert and Wolf's equations into computer code produced estimates that were valid up to thousands of samples before numerical errors corrupted the results. As the sample size was increased, the difference of increasing and increasingly similar parameter values caused the numerical accuracy to dramatically degrade. The authors have restructured Wolpert and Wolf's equations to get stable numerical results up to sample sizes in the billions. This report presents results from two steps in the process to produce numerically stable computer code that the authors believe can estimate information theoretic means and statistical errors for any sample size. This code is provided as a MATLAB software package available from MIT Lincoln Laboratory.

This page intentionally left blank.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

A problem that has plagued the tracking community for decades has been the lack of a single metric to assess the overall performance of tracking systems. The community has developed many different metrics and searched for methods to fuse the multiple measures into an overall score, with each selection of metrics and fusion method usually motivated by weakly supported arguments. Without an overall metric, it was difficult to determine if conflicting difference in performance for trackers, such as track length, track swaps, and track breaks, were leading to overall better or worse performance. For the intelligence, surveillance, and reconnaissance (ISR) community, the inability to recognize superior performance from trackers meant that it was difficult to acquire the right trackers for specific applications and to prove that performance was improving as trackers were tuned or new trackers acquired.

Research conducted by the authors over the last few years has identified a new technique to evaluate the performance of multi-target trackers [1]. The technique uses information theory to estimate the mutual information and conditional entropies for a set of truth tracks and a set of tracks generated by a tracking algorithm. The performance of the tracking algorithm is then assessed with the sum of the conditional entropies between the true tracks and those produced by the tracking algorithm. The technique is most effective for comparative studies of the relative performance of multiple trackers and classifiers. The technique is also useful as a way to optimize the parameters of individual trackers by identifying the parameter values that minimize the total conditional entropy. The technique is general enough that it can be used to evaluate classifiers and, by extension, any comparison where an N-to-M assignment can be made between a decision system and truth data [2].

The authors' prior publications neglected to consider the estimation of statistical errors for the information theoretic measures. The errors are needed to estimate the statistical significance of the results from the assessment of tracking, classification, and decision systems when algorithms are being compared with each other. A literature search for prior work on the estimation of information theoretic measures and error estimates turned up the papers of Wolpert and Wolf [3–5]. Wolpert and Wolf's papers provide equations to estimate statistical parameters from finite samples. They used a Bayesian analysis to derive closed form expressions for a number of statistical estimators, including an exact derivation of the first- and second-order statistics terms for the basic information theoretic entropy functions. The equations presented in the original papers were principally intended for the estimation of statistical parameters from small sample sizes. The equations are able to account for prior information in the statistics estimates, including various proposals for uninformative priors, including Haldane's prior (0), Perks' prior ($1/(n \times m)$) [6], Jeffrey's prior (1/2), and Bayes' or the uniform prior (1) [7]. The authors' information theoretic equations originally did not account for prior probabilities, and the adoption of Wolpert and Wolf's equations provide the means to account for prior probabilities.

In the course of implementing Wolpert and Wolf's equations for the numerical estimation of information theoretic measures with MATLAB, a few publication errors were found in the papers. After

correcting the equations, the authors' initial computer implementation was unexpectedly found to encounter machine precision problems around sample sizes of 10^7 samples. While this data sample size limit is more than sufficient for evaluating most classifiers, this number is potentially problematic for the evaluation of multi-target trackers. This is because the total number of samples for multi-target trackers is defined by the size of the state space that the trackers can potentially report. The sample size is usually defined by the number of cells in the state space, which is calculated from the product of the ratios of the range and the typical resolution of the measurements of each dimension. Sample sizes can easily range in the billions or higher for many data sets, with most of the counts assigned to the count of true negatives (an easily overlooked and often implied assumption in multi-target tracking is that a track is presumed to not exist if a tracker does not report it).

This report begins with a summary of Wolpert and Wolf's information theoretic functions, corrected for the typographical errors that appear in the papers. Performance results on the stability of a naïve coding of Wolpert and Wolf's are next presented to show that the numerical accuracy of the information theoretic variances and covariances becomes a severe problem for sample sizes on the order of a million. The report continues with two reformulations of the second-order statistics functions in an evolutionary process to produce software that estimates the information theoretic variances and covariances with better numerical accuracy. Analysis results on the numerical stability are presented for two of the series of reformulations. The first reformulation presented in this report produced software that was accurate up to the order of 10^{12} samples and the second reformulation up to the order of 10^{15} samples. Given computer code used double precision arithmetic with an accuracy of 10^{16}, the final reformulation is believed to be sufficiently precise that it will estimate reasonably accurate variances and covariances for all sample sizes. Examples are presented to show the use of the final computer code in the analysis of the relative performance of multi-target trackers and classifiers. The final section in the report provides a description of a software package that is available for download from MIT Lincoln Laboratory.

2. EVALUATION OF MULTI-TARGET TRACKERS AND CLASSIFIERS

The three most basic information measures for pairs of samples {x, y} are the entropies for the individual terms, $H(x)$ and $H(y)$, and the entropy for the combined terms, $H(x,y)$. From these, the mutual information between x and y can be calculated:

$$I(x,y) = H(x) + H(y) - H(x,y). \tag{1}$$

The two conditional entropies can be calculated from the relationships [8],

$$H(x \mid y) = H(x,y) - H(y), \text{ and} \tag{2}$$

$$H(y \mid x) = H(x,y) - H(x). \tag{3}$$

The performance score recommended by the authors is then given by the sum of the conditional entropies,

$$H(y \mid x) + H(x \mid y) = 2H(x,y) - H(x) - H(y). \tag{4}$$

Smaller values are better, with zero being the ideal performance point for a tracking system.

With a truth data set that can be processed by an algorithm, an accumulation matrix, r_{ij}, can be constructed that counts the number of times that subset i of the truth enumerations associates with a subset j of the output of the tracking or classification algorithm. The naïve approach to estimating entropy measures from counts is to convert the entries in the accumulation matrix into probabilities p_{ij} by normalizing the accumulation matrix with the sum of the total matrix, for example,

$$p_{ij} = r_{ij} \Big/ \sum_{i,j} r_{ij} \ . \tag{5}$$

The naïve probabilities, p_i and p_j, can also be calculated by summing the rows or columns of r_{ij} before using the pair of functions equivalent to Equation (5). The entropy can then be naïvely estimated with

$$H(x,y) = -\sum_{i,j} p_{ij} \ln(p_{ij}). \tag{6}$$

The other entropies, $H(x)$ and $H(y)$, can be estimated with equivalent functions based upon p_i and p_j. The conditional entropies and the total conditional entropy can be estimated by the appropriate combination of the three entropies. The advantage of the naïve approach is that the calculations of the mutual information terms are simply estimated from the accumulation matrix. However, this approach fails to account for prior probabilities and produces biased estimates when the sample size is small.

In addition to adopting total conditional entropy as the overall performance measure for tracker and classifier evaluation, the authors have defined terms to graphically show the performance of multi-target trackers and classifiers in a space that has similar characteristics to the receiver operation characteristic (ROC) plots. These plots show the probability of detection versus probability of false alarm or false alarm rate and are often used to visualize the performance of detection algorithms. The (1) ratio of mutual information to truth entropy and the (2) conditional information of the sensor data given the truth divided by the truth entropy are used as the y- and x-axes for the graphs. The y-axis is referred to as information completeness and the x-axis is referred to as the false information ratio. The y-axis values range from zero to one, while the x-axis values are non-negative with no definite upper limit. The normalization by the truth entropy is used so that the y-axis values depict the fractional amount of true information reported by the decision algorithm. The plotted values are fairly stable against different approaches to estimating the information theoretic measures. This sensitivity is examined in detail in Section 8. Figure 1 shows an example of the information measures for two simulated tracker data sets. The authors refer to the plot as an information coverage plot. The dotted diagonal lines are isobars for the total conditional entropy divided by the total truth entropy. The sum of information completeness plus the ratio of the conditional entropy of the truth data given the sensor to the truth entropy is one. The ideal sensor produces zero total condition entropy, which would be plotted at the position (0,1) in Figure 1. The reader could obtain a sense of the significance of the differences between the two trackers in Figure 1 if error ellipses were plotted. For this simulated data set, the total conditional entropy for Tracker 1 is 0.0142 and for Tracker 2 is 0.0216. Tracker 1 is better than Tracker 2, but without error estimates it is difficult to determine the significance of the test results.

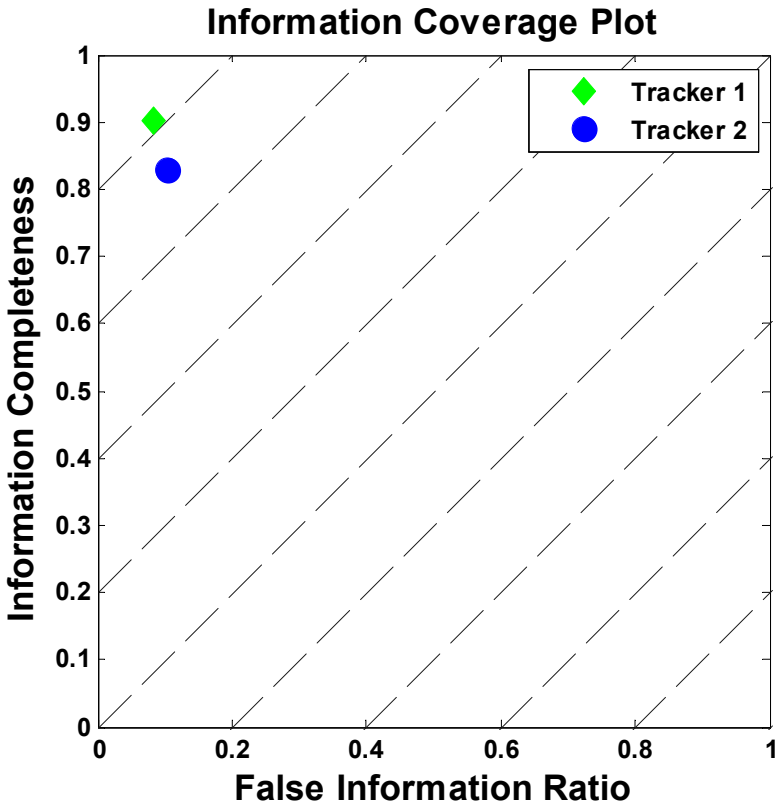


Figure 1. An example of an information coverage plot based upon data from two simulated trackers.

If means, μ, and standard deviations, σ, for the total conditional entropy can be calculated for multiple tracking or classification systems under test, a rough statistical test can be performed to assess the reliability of the decision that one system performs better than another. If it is assumed that the probability distribution functions for the true values for the total conditional entropy of the trackers can be described by Gaussian functions centered on the estimated means and with spreads characterized by the standard deviations, then a reasonable test statistic is

$$P\left(x_{2T} < x_{1T}\right) = \frac{1}{2\pi\sigma_1\sigma_2}\int_{-\infty}^{\infty}\int_{-\infty}^{x_1}\exp\left(-\frac{\left(x_1-\mu_1\right)^2}{2\sigma_1^2}\right)\exp\left(-\frac{\left(x_2-\mu_2\right)^2}{2\sigma_2^2}\right)dx_1dx_2\ . \tag{7}$$

This integral is the probability that the true (and unknown) value of the total conditional entropy for Tracker 2, x_{2T}, is less than the true value for Tracker 1, x_{1T}. The test is a rough estimate because the actual information theoretic probability distributions are not Gaussian and the information theoretic

measures are always non-negative values. Even with these limitations, Equation (7) can still provide useful estimates of the statistical significance of the results from comparisons of trackers and classifiers.

For actual use in calculations, the integrals in Equation (7) can be simplified with the integral [9]

$$\int_{-\infty}^{\infty} G'(x)G(a+bx)dx = G\left(\frac{a}{\sqrt{1+b^2}}\right), \tag{8}$$

where

$$G'(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right) \tag{9}$$

and

$$G(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{t^2}{2}\right)dt\,. \tag{10}$$

The result is that Equation (7) can be reduced to

$$T = G\left(\frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right). \tag{11}$$

The remainder of the report works through the effort to develop computer code to estimate means and standard deviations so that the statistical significance of information theoretic evaluations of trackers and classifiers can be estimated for any test data set.

3. DERIVATIONS FOR COMPUTER CALCULATIONS

The papers of Wolpert and Wolf provide exact formulae for estimating a number of first- and second-order statistics terms from which means and standard deviation can be obtained. The primary information theoretic derivations of interest for this report are the joint entropy and the two independent entropies for two quantized variables, x and y, as well as the variance and covariance terms that are associated with the three entropies of Equations (1) through (3). Given estimates for the three entropies and associated error terms, the values for mutual information and conditional entropies and associated error terms can be calculated. It should be noted that the original papers of Wolpert and Wolf contained a few typographical errors that have been corrected here. A summary of the corrected results of their derivations are provided below.

3.1 CALCULATION OF THE ENTROPY MEASURES

The relationship between Wolpert and Wolf's variables and the expectation values for the three entropies are

$$E\left(H\left(x,y\right)\right) = -\mathcal{E}_{\overline{IJ}}, \tag{12}$$

$$E\left(H\left(x\right)\right) = -\mathcal{E}_{\overline{I}}, \text{ and} \tag{13}$$

$$E\left(H\left(y\right)\right) = -\mathcal{E}_{\overline{J}}. \tag{14}$$

The variables, $\mathcal{E}_{\overline{IJ}}$, $\mathcal{E}_{\overline{I}}$ and $\mathcal{E}_{\overline{J}}$, in Equations (12) through (14) are derived by Wolpert and Wolf to be

$$\mathcal{E}_{\overline{IJ}} = \sum_{i,j} \frac{\nu_{ij}}{\nu} \Delta\Phi^{(1)}\left(\nu_{ij}+1, \nu+1\right), \tag{15}$$

$$\mathcal{E}_{\overline{I}} = \sum_{i} \frac{\nu_{i\bullet}}{\nu} \Delta\Phi^{(1)}\left(\nu_{i\bullet}+1, \nu+1\right), \text{ and} \tag{16}$$

$$\mathcal{E}_{\overline{J}} = \sum_{j} \frac{\nu_{\bullet j}}{\nu} \Delta\Phi^{(1)}\left(\nu_{\bullet j}+1, \nu+1\right). \tag{17}$$

The variables in Equations (15) through (17) are defined as follows: The measurements of interest for the estimation of the entropies are the prior adjusted sample size ν, the prior adjusted number of co-occurrences of event i and event j, ν_{ij}, the prior adjusted number of occurrences of event i, $\nu_{i\bullet}$, and of event j, $\nu_{\bullet j}$. Wolpert and Wolf considered a number of different forms for prior probabilities in their

Bayesian derivation of statistical measures. In this report, the forms of interest for the prior probabilities is restricted to the consideration of those that are of the form

$$P(\mathrm{p}) \propto \Delta\,\mathrm{p} \prod_{i=1}^{m} p_i^{r_i} \,. \tag{18}$$

The Dirichlet prior is a special case of Equation (18) where all prior constants, r_i, are the same value, r. Combining the sample counts with the prior counts produces the prior adjusted counts for the variables in Equations (15) through (17):

$$\nu = n + \sum_i \sum_j r_{ij} \,, \tag{19}$$

$$\nu_{ij} = n_{ij} + r_{ij} \,, \tag{20}$$

$$\nu_{i\bullet} = n_{i\bullet} + r_{i\bullet} \,, \text{ and} \tag{21}$$

$$\nu_{\bullet j} = n_{\bullet j} + r_{\bullet j} \,, \tag{22}$$

where n is the total sample size, $n_{i\bullet}$ is the count of samples in class i, $n_{\bullet j}$ is the count of samples in class j, and n_{ij} is the count of samples in class i and j.

Hypergeometric functions, $\Gamma(z)$ (or equivalently named, gamma functions), appear in a number of the formulae. The digamma function, $\Psi^{(0)}(z)$, is defined to be the first derivative of the log of the gamma function. For any power of derivatives, the generalization of the digamma function is

$$\Psi^{(n)}(z) = \partial_z^{n+1} \ln(\Gamma(z)). \tag{23}$$

In this notation, n is one less than the order of differentiation, as defined in conventional practice. Wolpert and Wolf defined the following function to simplify the representation of terms in their equations:

$$\Phi^{(n)}(z_1) = \Psi^{(n-1)}(z). \tag{24}$$

The function $\Delta\Phi^{(n)}(z_1, z_2)$ was thus defined to be

$$\Delta\Phi^{(n)}(z_1, z_2) = \Phi^{(n)}(z_1) - \Phi^{(n)}(z_2), \tag{25}$$

and is the last of the definitions needed for Equations (15) through (17).

The second-order statistics for the three fundamental entropies, $H(x,y)$, $H(x)$, and $H(y)$, require six functions. Two pairs of the functions are symmetric under interchange of x and y, so only four functions need to be explicitly provided. The two additional functions are obtained by exchanging x and y variables.

The second-order statistics term for $E\left(H(x,y)H(x,y)\right)$, or $\mathcal{E}_{\overline{IJMN}}$, as defined by Wolpert and Wolf, is

$$\begin{aligned}\mathcal{E}_{\overline{IJMN}} = &\sum_{i,j}\sum_{m,n\neq i,j}\frac{\nu_{ij}\nu_{mn}}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{mn}+1,\nu+2)-\Phi^{(2)}(\nu+2)\right\}\\ &+\sum_{i,j}\frac{\nu_{ij}(\nu_{ij}+1)}{\nu(\nu+1)}\left\{\left[\Delta\Phi^{(1)}(\nu_{ij}+2,\nu+2)\right]^2+\Delta\Phi^{(2)}(\nu_{ij}+2,\nu+2)\right\}.\end{aligned} \tag{26}$$

The second-order statistic term for $E\left(H(x,y)H(x)\right)$, or $\mathcal{E}_{\overline{IJM}}$, is

$$\begin{aligned}\mathcal{E}_{\overline{IJM}} = &\sum_{i,j}\sum_{m\neq i}\frac{\nu_{ij}\nu_{m\bullet}}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+2)-\Phi^{(2)}(\nu+2)\right\}\\ &+\sum_{i,j}\frac{\nu_{ij}(\nu_{i\bullet}+1)}{\nu(\nu+1)}\left\{\left[\Delta\Phi^{(1)}(\nu_{i\bullet}+2,\nu+2)\right]^2\right.\\ &\left.+\Delta\Phi^{(1)}(\nu_{ij}+1,\nu_{i\bullet}+1)\Delta\Phi^{(1)}(\nu_{i\bullet}+2,\nu+2)+\Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2)\right\}.\end{aligned} \tag{27}$$

The second-order statistic for $E\left(H(x,y)H(y)\right)$, or $\mathcal{E}_{\overline{IJN}}$, can be obtained by substituting $\bullet n$ for $m\bullet$, and n for m, with the summation modified to be over n, excluding j instead of i.

The second-order statistic term for $E\left(H(x)H(x)\right)$, or $\mathcal{E}_{\overline{IM}}$, is

$$\begin{aligned}\mathcal{E}_{\overline{IM}} = &\sum_{i,m\neq i}\frac{\nu_{i\bullet}\nu_{m\bullet}}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+2)-\Phi^{(2)}(\nu+2)\right\}\\ &+\sum_{i}\frac{\nu_{i\bullet}(\nu_{i\bullet}+1)}{\nu(\nu+1)}\left\{\left[\Delta\Phi^{(1)}(\nu_{i\bullet}+2,\nu+2)\right]^2+\Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2)\right\}.\end{aligned} \tag{28}$$

The second-order statistic for $E\left(H(y)H(y)\right)$, or $\mathcal{E}_{\overline{JN}}$, can be obtained by substituting $i\bullet$ for $\bullet j$, $m\bullet$ for $\bullet n$, and swapping i for j and n for m.

The second-order statistic term for $E\left(H(x)H(y)\right)$, or $\mathcal{E}_{\overline{IN}}$, is the most complicated of the four expressions, partly because it contains terms with infinite sums:

$$
\begin{aligned}
\mathcal{E}_{\overline{IN}} = \sum_{i,n} \frac{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}{\nu\left(\nu+1\right)} \Bigg\{ & \left\{ \left[\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right]^2 + \Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right) \right\} \\
& \times \left(1 - \frac{\left(\nu_{i\bullet}-\nu_{in}\right)+\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}} + \frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)} \right) \\
& + \Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right) \times \\
& \sum_{r=0}^{\infty} \frac{Q_1(r,1)}{r!} \left[\frac{\left(\nu_{\bullet n}-\nu_{in}\right)_r}{\left(\overline{\nu}_{in}\right)_r} \left(1 - \frac{\nu_{i\bullet}-\nu_{in}}{\overline{\nu}_{in}+r} \right) + \frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r}{\left(\overline{\nu}_{in}\right)_r} \left(1 - \frac{\nu_{\bullet n}-\nu_{in}}{\overline{\nu}_{in}+r} \right) \right] \\
& + \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} \frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r\left(\nu_{\bullet n}-\nu_{in}\right)_s}{\left(\overline{\nu}_{in}\right)_{r+s}} \frac{Q_1(r,1)}{r!}\frac{Q_1(s,1)}{s!} \Bigg\} \ .
\end{aligned}
\tag{29}
$$

This equation is a corrected for typographical errors in the papers of Wolpert and Wolf.

The variables and nomenclature in Equation (29) that have not been previously defined are

$$\overline{\nu}_{in} = \nu_{i\bullet} + \nu_{\bullet n} - \nu_{in} \text{ and} \tag{30}$$

$$\left(a\right)_b = \Gamma\left(a+b\right)/\Gamma\left(a\right), \tag{31}$$

where $\left(a\right)_b$ is Pochhammer's symbol [10] and is used to represent the ratio of two gamma functions. In this usage, it is a rising factorial as opposed to a falling factorial.

Lastly,

$$Q_1\left(j,\eta_1\right) = \left[1-\Theta\left(j-\eta_1-1\right)\right]\frac{\left(-1\right)^j}{\left(\eta_1-j\right)!}\sum_{r=0}^{j-1}\frac{1}{\eta_1-r} + \Theta\left(j-\eta_1-1\right)\left(-1\right)^{\eta_1+1}\Gamma\left(j-\eta_1\right). \tag{32}$$

The Heaviside theta function in Equation (32) is defined as

$$\Theta\left(x\ge 0\right)=1, \quad \Theta\left(x<0\right)=0\,. \tag{33}$$

The expansion of the Q_1 function is necessary for eventual implementation of $\mathcal{E}_{\overline{IN}}$ in computer software. For $\mathcal{E}_{\overline{IN}}$, the $Q_1\left(j,\eta_1\right)$ functions are only evaluated for $\eta_1 = 1$. This limits the functions of interest to

$$Q_1\left(j,1\right) = \left[1-\Theta\left(j-2\right)\right]\frac{\left(-1\right)^j}{\left(1-j\right)!}\left[\sum_{r=0}^{j-1}\frac{1}{1-r}\right] + \Theta\left(j-2\right)\Gamma\left(j-1\right)\ . \tag{34}$$

By standard usage, the sum is defined to be zero for upper limits less than lower limits. The three classes of interest for the expansion of the Q_1 function are

$$Q_1(0,1)=0; \quad j=0, \tag{35}$$

$$Q_1(1,1)=-1; \quad j=1, \text{ and} \tag{36}$$

$$Q_1(j,1)=\Gamma(j-1); \quad j\geq 2. \tag{37}$$

Thus, the $j=0$ terms in all infinite sums can be ignored. Because the Q_1 terms are always associated with division by a gamma function, the following simplifications are useful:

$$\frac{Q_1(1,1)}{1!}=-1; \quad j=1, \tag{38}$$

$$\frac{Q_1(j,1)}{j!}=\frac{\Gamma(j-1)}{j!}=\frac{(j-2)!}{j!}=\frac{1}{j(j-1)}; \quad j\geq 2. \tag{39}$$

The single-term infinite sums can be redefined as infinite sums of the form,

$$S_1(\nu_x,\nu_y,\nu_{in},\bar{\nu}_{in})=\sum_{r=0}^{\infty}\frac{Q_1(r,1)}{r!}\left[\frac{(\nu_x-\nu_{in})_r}{(\bar{\nu}_{in})_r}\left(1-\frac{\nu_y-\nu_{in}}{\bar{\nu}_{in}+r}\right)\right], \tag{40}$$

and rewritten as

$$\begin{aligned} S_1(\nu_x,\nu_y,\nu_{in},\bar{\nu}_{in})=&-\frac{(\nu_x-\nu_{in})_1}{(\bar{\nu}_{in})_1}\left(1-\frac{\nu_y-\nu_{in}}{\bar{\nu}_{in}+1}\right)\\ &+\sum_{r=2}^{\infty}\frac{1}{r(r-1)}\left[\frac{(\nu_x-\nu_{in})_r}{(\bar{\nu}_{in})_r}\left(1-\frac{\nu_y-\nu_{in}}{\bar{\nu}_{in}+r}\right)\right]. \end{aligned} \tag{41}$$

With the use of Equation (31), the function can be written as

$$\begin{aligned} S_1(\nu_x,\nu_y,\nu_{in},\bar{\nu}_{in})=&-\frac{\Gamma(\nu_x-\nu_{in}+1)\Gamma(\bar{\nu}_{in})}{\Gamma(\nu_x-\nu_{in})\Gamma(\bar{\nu}_{in}+1)}\left(1-\frac{\nu_y-\nu_{in}}{\bar{\nu}_{in}+1}\right)\\ &+\sum_{r=2}^{\infty}\frac{1}{r(r-1)}\left[\frac{(\nu_x-\nu_{in})_r}{(\bar{\nu}_{in})_r}\left(1-\frac{\nu_y-\nu_{in}}{\bar{\nu}_{in}+r}\right)\right]. \end{aligned} \tag{42}$$

The use of the recursion relationship

$$\Gamma(a+1) = a\Gamma(a) \tag{43}$$

simplifies the function to

$$\begin{aligned} S_1(\nu_x, \nu_y, \nu_{in}, \overline{\nu}_{in}) = & -\frac{(\nu_x - \nu_{in})}{\overline{\nu}_{in}}\left(1 - \frac{\nu_y - \nu_{in}}{\overline{\nu}_{in}+1}\right) \\ & + \sum_{r=2}^{\infty} \frac{1}{r(r-1)}\left[\frac{(\nu_x - \nu_{in})_r}{(\overline{\nu}_{in})_r}\left(1 - \frac{\nu_y - \nu_{in}}{\overline{\nu}_{in}+r}\right)\right]. \end{aligned} \tag{44}$$

Equation (30) can be used to get

$$\begin{aligned} S_1(\nu_x, \nu_y, \nu_{in}, \overline{\nu}_{in}) = & -\frac{(\nu_x - \nu_{in})(\nu_x+1)}{\overline{\nu}_{in}(\overline{\nu}_{in}+1)} \\ & + \sum_{r=2}^{\infty} \frac{1}{r(r-1)}\left[\frac{(\nu_x - \nu_{in})_r}{(\overline{\nu}_{in})_r}\left(1 - \frac{\nu_y - \nu_{in}}{\overline{\nu}_{in}+r}\right)\right]. \end{aligned} \tag{45}$$

The double summation can be redefined with the function

$$S_2(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}, \overline{\nu}_{in}) = \sum_{r=0}^{\infty}\sum_{s=0}^{\infty} \frac{(\nu_{i\bullet} - \nu_{in})_r (\nu_{\bullet n} - \nu_{in})_s}{(\overline{\nu}_{in})_{r+s}} \frac{Q_1(r,1)}{r!} \frac{Q_1(s,1)}{s!} \tag{46}$$

and expanded to

$$\begin{aligned} S_2(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}, \overline{\nu}_{in}) = & \frac{(\nu_{i\bullet} - \nu_{in})_1 (\nu_{\bullet n} - \nu_{in})_1}{(\overline{\nu}_{in})_2} \\ & - \sum_{s=2}^{\infty} \frac{(\nu_{i\bullet} - \nu_{in})_1 (\nu_{\bullet n} - \nu_{in})_s}{(\overline{\nu}_{in})_{1+s}} \frac{Q_1(s,1)}{s!} \\ & - \sum_{r=2}^{\infty} \frac{(\nu_{i\bullet} - \nu_{in})_r (\nu_{\bullet n} - \nu_{in})_1}{(\overline{\nu}_{in})_{r+1}} \frac{Q_1(r,1)}{r!} \\ & + \sum_{r=2}^{\infty}\sum_{s=2}^{\infty} \frac{(\nu_{i\bullet} - \nu_{in})_r (\nu_{\bullet n} - \nu_{in})_s}{(\overline{\nu}_{in})_{r+s}} \frac{Q_1(r,1)}{r!} \frac{Q_1(s,1)}{s!}, \end{aligned} \tag{47}$$

as was done for the single sums.

Using a similar approach to that used for the equations of S_1, Equation (47) can be rewritten as

$$\begin{aligned} S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right) &= \frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\left(\overline{\nu}_{in}\right)_2} \\ &\quad -\left(\nu_{i\bullet}-\nu_{in}\right)\sum_{s=2}^{\infty}\frac{\left(\nu_{\bullet n}-\nu_{in}\right)_s}{\left(\overline{\nu}_{in}\right)_{1+s}}\frac{1}{s(s-1)} \\ &\quad -\left(\nu_{\bullet n}-\nu_{in}\right)\sum_{r=2}^{\infty}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r}{\left(\overline{\nu}_{in}\right)_{r+1}}\frac{1}{r(r-1)} \\ &\quad +\sum_{r=2}^{\infty}\sum_{s=2}^{\infty}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r\left(\nu_{\bullet n}-\nu_{in}\right)_s}{\left(\overline{\nu}_{in}\right)_{r+s}}\frac{1}{r(r-1)}\frac{1}{s(s-1)}, \end{aligned} \tag{48}$$

which means that $\mathcal{E}_{\overline{IN}}$ can be rewritten as

$$\begin{aligned} \mathcal{E}_{\overline{IN}} &= \sum_{i,n}\frac{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}{\nu(\nu+1)}\left\{\left\{\left[\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right]^2+\Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right\}\right. \\ &\quad \times\left(1-\frac{\left(\nu_{i\bullet}-\nu_{in}\right)+\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}}+\frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}\right) \\ &\quad +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\left[S_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right)+S_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in},\overline{\nu}_{in}\right)\right] \\ &\quad \left.+S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right)\right\}. \end{aligned} \tag{49}$$

The above function was coded in MATLAB to produce the first version of an algorithm to numerically estimate the means and covariances of the three entropy measures.

3.2 CALCULATION OF ADDITIONAL INFORMATION THEORETIC MEASURES

The mutual information and conditional entropies can be estimated from the three entropy values, $H(x,y)$, $H(x)$, and $H(y)$. The mutual information is given by

$$I(x,y)=H(x)+H(y)-H(x,y)=\mathcal{E}_{\overline{IJ}}-\mathcal{E}_{\overline{I}}-\mathcal{E}_{\overline{J}}\ . \tag{50}$$

The two conditional entropies are given by

$$H(x\mid y)=H(x,y)-H(y)=\mathcal{E}_{\overline{J}}-\mathcal{E}_{\overline{IJ}}\ \text{and} \tag{51}$$

$$H(y\mid x)=H(x,y)-H(x)=\mathcal{E}_{\overline{I}}-\mathcal{E}_{\overline{IJ}}\ . \tag{52}$$

The sum of the conditional entropies is used by the authors as an overall score for trackers and classifiers,

$$H(x \mid y) + H(y \mid x) = 2H(x,y) - H(y) - H(x) = \mathcal{E}_{\overline{I}} + \mathcal{E}_{\overline{J}} - 2\mathcal{E}_{\overline{IJ}}. \tag{53}$$

The second moment terms for these additional variables are shown in Table 1 and are needed to estimate the variances and covariances for all six information theoretic measures.

TABLE 1

A Table of Second-Order Statistics Functions for Common Information Theoretic Variables

Terms	$I(x,y)$	$H(x \mid y)$	$H(y \mid x)$	$H(x \mid y) + H(y \mid x)$
$H(x,y)$	$\mathcal{E}_{\overline{IJM}} + \mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{IJMN}}$	$\mathcal{E}_{\overline{IJMN}} - \mathcal{E}_{\overline{IJN}}$	$\mathcal{E}_{\overline{IJMN}} - \mathcal{E}_{\overline{IJM}}$	$2\mathcal{E}_{\overline{IJMN}} - \mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{IJM}}$
$H(x)$	$\mathcal{E}_{\overline{IM}} + \mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{IJM}}$	$\mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IN}}$	$\mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IM}}$	$2\mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{IM}}$
$H(y)$	$\mathcal{E}_{\overline{IN}} + \mathcal{E}_{\overline{JN}} - \mathcal{E}_{\overline{IJN}}$	$\mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{JN}}$	$\mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{IN}}$	$2\mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{JN}} - \mathcal{E}_{\overline{IN}}$
$I(x,y)$	$\mathcal{E}_{\overline{IJMN}} + \mathcal{E}_{\overline{IM}} + \mathcal{E}_{\overline{JN}}$ $-2\left(\mathcal{E}_{\overline{IJM}} + \mathcal{E}_{\overline{IJN}} - \mathcal{E}_{\overline{IN}}\right)$	$-\mathcal{E}_{\overline{IJMN}} + 2\mathcal{E}_{\overline{IJN}}$ $+\mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{JN}}$	$-\mathcal{E}_{\overline{IJMN}} + 2\mathcal{E}_{\overline{IJM}} + \mathcal{E}_{\overline{IJN}}$ $-\mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{IM}}$	$-2\mathcal{E}_{\overline{IJMN}} + 3\mathcal{E}_{\overline{IJN}} + 3\mathcal{E}_{\overline{IJM}}$ $-2\mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{JN}} - \mathcal{E}_{\overline{IM}}$
$H(x \mid y)$		$\mathcal{E}_{\overline{IJMN}} + \mathcal{E}_{\overline{JN}} - 2\mathcal{E}_{\overline{IJN}}$	$\mathcal{E}_{\overline{IJMN}} - \mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IJN}} + \mathcal{E}_{\overline{IN}}$	$2\mathcal{E}_{\overline{IJMN}} + \mathcal{E}_{\overline{JN}} - 3\mathcal{E}_{\overline{IJN}}$ $-\mathcal{E}_{\overline{IJM}} + \mathcal{E}_{\overline{IN}}$
$H(y \mid x)$			$\mathcal{E}_{\overline{IJMN}} + \mathcal{E}_{\overline{IM}} - 2\mathcal{E}_{\overline{IJM}}$	$2\mathcal{E}_{\overline{IJMN}} + \mathcal{E}_{\overline{IM}} - 3\mathcal{E}_{\overline{IJM}}$ $-\mathcal{E}_{\overline{IJN}} + \mathcal{E}_{\overline{IN}}$
$H(x \mid y) + H(y \mid x)$				$4\mathcal{E}_{\overline{IJMN}} - 4\mathcal{E}_{\overline{IJN}} - 4\mathcal{E}_{\overline{IJM}}$ $+\mathcal{E}_{\overline{IM}} + 2\mathcal{E}_{\overline{IN}} + \mathcal{E}_{\overline{JN}}$

4. NUMERICAL RESULTS FROM THE INITIAL ALGORITHM

To evaluate the performance of the initial algorithm, a test accumulation or confusion matrix was used as input to the function. The accumulation matrix that was chosen for numerical evaluation was

$$C = \begin{pmatrix} .47 & .03 & .04 & .00 & .16 & .12 & .11 & .08 \\ .03 & .58 & .07 & .08 & .15 & .03 & .05 & .00 \\ .10 & .00 & .41 & .14 & .19 & .10 & .06 & .00 \\ .08 & .05 & .12 & .50 & .05 & .12 & .05 & .03 \\ .08 & .00 & .07 & .08 & .47 & .26 & .04 & .00 \\ .01 & .00 & .11 & .10 & .15 & .48 & .08 & .07 \\ .18 & .00 & .23 & .03 & .13 & .00 & .38 & .08 \\ .13 & .00 & .03 & .02 & .18 & .07 & .12 & .45 \end{pmatrix} \tag{54}$$

and comes from a paper by Farhadi et al. [11]. For the evaluation here, the matrix was scaled by different factors to evaluate the accuracy of the resultant means and covariances for different simulated sample sizes. Figure 2 shows how the covariance estimates change with sample size. Negative eigenvalues are indicated by the red circles. It can be seen that one of the eigenvalues in the covariance matrix becomes negative above 10^7, which is clearly incorrect because all eigenvalues are required to be positive for covariance matrices. Rounding errors are so severe at these sample sizes that negative eigenvalues are calculated. Greater numerical precision in the calculations is needed to obtain accurate estimates from the calculations (Hints that there were publication errors in the Wolpert and Wolf papers originally came from examining the eigenvalues of the covariance matrices for the three entropies). The correct trend would be for the eigenvalues to continue to decrease approximately log-linearly as the sample size increases. The term that suffered the most from numerical precision problems in the initial coding of the algorithm was $\mathcal{E}_{\overline{IN}}$.

The terminology for this report is to use the phrase ‘accumulation matrix’ to refer to matrices that are sometimes called ‘confusion matrices’ in the context of the evaluation of classifiers. The phrase ‘association matrix’ is used to refer to matrices constructed to determine the optimal association between sets of tracks. The association matrices are used to optimally associate truth and estimated tracks at individual frames or time steps. The accumulation matrix is used to count the number of times that pairs of tracks optimally associate between the two track data sets.

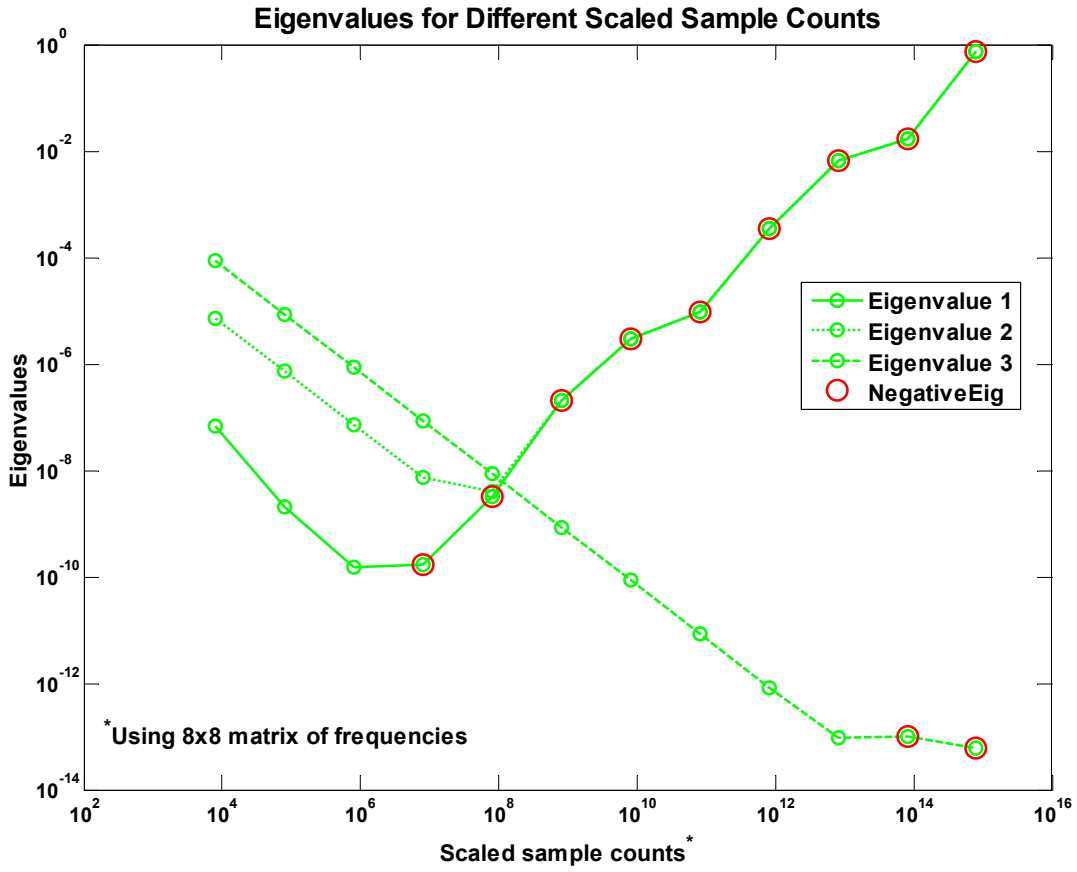


Figure 2. The absolute values of eigenvalues from the covariance matrices of entropies for different scale factors of the accumulation matrix in Equation (54) using the initial algorithm. The x-axis is the scaled sample counts in the matrix. Negative eigenvalues are shown with red circles to indicate where the covariance terms are incorrectly estimated.

5. DERIVING COVARIANCE EQUATIONS FROM FIRST- AND SECOND-ORDER EQUATIONS

There were a number of causes for the limited precision in the numerical estimates from the first algorithm. First, the variance and covariance terms were numerically calculated from the difference between second-order statistics terms and squares of first-order statistics terms. In general, as sample sizes increase, the second-order statistics terms converge toward the square of the first-order statistics terms because errors decrease as sample sizes increase. The numerical estimates of the variances and covariances become more and more inaccurate as the algorithm calculates differences between pairs of increasingly large and similar numbers. Analytically deriving the covariance formulae from the difference between second- and first-order statistics terms and coding that resultant function was the first step used to make the calculations less sensitive to machine precision errors. This reformulation produced an algorithm with equations that accounted for most of the cancellation between terms before any numerical calculations are performed.

A second technique used to reduce machine precision errors was to identify pairs of terms in individual variance and covariance equations that nearly cancel and arrange the calculations to account for this cancellation. The estimates of the infinite sums involved in the calculation of $\mathfrak{S}_{\overline{IN}}$ improved the most from this approach.

5.1 DERIVATION OF THE FIRST FIVE COVARIANCE TERMS

Five of the six statistics terms are relatively straightforward to analytically reformulate in terms of variance and covariance terms; the sixth term is much more complicated. Derivations for the first five terms are presented in the section before presenting the derivation of the sixth formula in its own section.

The covariance term for $H(x,y)$ is given by

$$\sigma^2_{\overline{IJMN}} = E\big(H(x,y)H(x,y)\big) - E\big(H(x,y)\big)^2 = \mathfrak{S}_{\overline{IJMN}} - \mathfrak{S}_{\overline{IJ}}\mathfrak{S}_{\overline{MN}} \tag{55}$$

and expands to

$$\begin{aligned}\sigma^2_{\overline{IJMN}} = &\sum_{i,j}\sum_{m,n\neq i,j}\frac{\nu_{ij}\nu_{mn}}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+2\right)\Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+2\right)-\Phi^{(2)}\left(\nu+2\right)\right\}\\ &+\sum_{i,j}\frac{\nu_{ij}\left(\nu_{ij}+1\right)}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}\left(\nu_{ij}+2,\nu+2\right)^2+\Delta\Phi^{(2)}\left(\nu_{ij}+2,\nu+2\right)\right\}\\ &-\sum_{i,j}\frac{\nu_{ij}}{\nu}\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right)\sum_{m,n}\frac{\nu_{mn}}{\nu}\Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+1\right).\end{aligned} \tag{56}$$

It is convenient to treat $\sigma^2_{\overline{IJMN}}$ as a combination of diagonal and off-diagonal components,

$$\sigma^2_{\overline{IJMN}} = {}_D\sigma^2_{\overline{IJMN}} + {}_{OD}\sigma^2_{\overline{IJMN}}, \tag{57}$$

and rewrite Equation (56) as the pair of equations,

$$\begin{aligned} {}_D\sigma^2_{\overline{IJMN}} = &+\sum_{i,j} \frac{\nu_{ij}\left(\nu_{ij}+1\right)}{\nu\left(\nu+1\right)} \left\{\Delta\Phi^{(1)}\left(\nu_{ij}+2,\nu+2\right)^2 + \Delta\Phi^{(2)}\left(\nu_{ij}+2,\nu+2\right)\right\} \\ &-\sum_{i,j}\left(\frac{\nu_{ij}}{\nu}\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right)\right)^2 \end{aligned} \tag{58}$$

and

$$\begin{aligned} {}_{OD}\sigma^2_{\overline{IJMN}} = &\sum_{i,j}\sum_{m,n\neq i,j} \frac{\nu_{ij}\nu_{mn}}{\nu\left(\nu+1\right)} \left\{\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+2\right)\Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+2\right) - \Phi^{(2)}\left(\nu+2\right)\right\} \\ &-\sum_{i,j}\frac{\nu_{ij}}{\nu}\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right)\sum_{m,n\neq i,j}\frac{\nu_{mn}}{\nu}\Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+1\right) . \end{aligned} \tag{59}$$

The next step is to modify the $\Delta\Phi^{(1)}$ terms to have the same parameter values for as many subsets of terms as possible so as to get the maximum cancellation between terms. The first-order statistics functions contain terms with $\nu-1$, whereas the second-order statistics functions contain terms with $\nu-2$. The following recursion relation [12] is useful for converting the functions to consistent parameter values:

$$\Phi^{(1)}\left(z+1\right) = \Psi_0\left(z+1\right) = \Psi_0\left(z\right) + \frac{1}{z} \ , \tag{60}$$

which means that

$$\Delta\Phi^{(1)}\left(z_1+1,z_2+2\right) = \Delta\Phi^{(1)}\left(z_1+1,z_2+1\right) - \frac{1}{z_2+1} \tag{61}$$

and

$$\Delta\Phi^{(1)}\left(z_1+2,z_2+2\right) = \Delta\Phi^{(1)}\left(z_1+1,z_2+1\right) + \frac{1}{z_1+1} - \frac{1}{z_2+1} . \tag{62}$$

Working through the substitutions and the algebraic manipulations results in

$$
{}_{D}\sigma^{2}_{\overline{IJMN}} = \sum_{i,j} \frac{\nu_{ij}\left(\nu_{ij}+1\right)}{\nu\left(\nu+1\right)} \left[\frac{\nu-\nu_{ij}}{\left(\nu_{ij}+1\right)\nu} \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right)^{2} \right.
$$

$$
\left. + 2\frac{\nu-\nu_{ij}}{\left(\nu_{ij}+1\right)\left(\nu+1\right)} \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right) + \left(\frac{\nu-\nu_{ij}}{\left(\nu_{ij}+1\right)\left(\nu+1\right)} \right)^{2} + \Delta\Phi^{(2)}\left(\nu_{ij}+2,\nu+2\right) \right] \tag{63}
$$

and

$$
{}_{OD}\sigma^{2}_{\overline{IJMN}} = \sum_{i,j} \sum_{m,n\neq i,j} \frac{\nu_{ij}\nu_{mn}}{\nu\left(\nu+1\right)} \left[-\frac{1}{\nu} \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right) \Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+1\right) \right.
$$

$$
\left. + \frac{1}{\nu+1} \left(\frac{1}{\nu+1} - \left(\Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right) + \Delta\Phi^{(1)}\left(\nu_{mn}+1,\nu+1\right) \right) \right) - \Phi^{(2)}\left(\nu+2\right) \right] . \tag{64}
$$

The same can be done for the next term,

$$
\begin{aligned}
\sigma^{2}_{\overline{IJM}} &= \mathcal{E}_{\overline{IJM}} - \mathcal{E}_{\overline{IJ}} \mathcal{E}_{\overline{M}} \\
&= \sum_{i,j} \sum_{m\neq i} \frac{\nu_{ij}\nu_{m\bullet}}{\nu\left(\nu+1\right)} \left\{ \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+2\right) \Delta\Phi^{(1)}\left(\nu_{m\bullet}+1,\nu+2\right) - \Phi^{(2)}\left(\nu+2\right) \right\} \\
&\quad + \sum_{i,j} \frac{\nu_{ij}\left(\nu_{i\bullet}+1\right)}{\nu\left(\nu+1\right)} \left\{ \left[\Delta\Phi^{(1)}\left(\nu_{i\bullet}+2,\nu+2\right) \right]^{2} \right. \\
&\quad \left. + \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu_{i\bullet}+1\right) \Delta\Phi^{(1)}\left(\nu_{i\bullet}+2,\nu+2\right) + \Delta\Phi^{(2)}\left(\nu_{i\bullet}+2,\nu+2\right) \right\} \\
&\quad - \sum_{i,j} \frac{\nu_{ij}}{\nu} \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right) \sum_{m} \frac{\nu_{m\bullet}}{\nu} \Delta\Phi^{(1)}\left(\nu_{m\bullet}+1,\nu+1\right) ,
\end{aligned} \tag{65}
$$

which can again be split into diagonal and off-diagonal terms,

$$
\sigma^{2}_{\overline{IJM}} = {}_{D}\sigma^{2}_{\overline{IJM}} + {}_{OD}\sigma^{2}_{\overline{IJM}} , \tag{66}
$$

$$
\begin{aligned}
{}_{D}\sigma^{2}_{\overline{IJM}} &= + \sum_{i,j} \frac{\nu_{ij}\left(\nu_{i\bullet}+1\right)}{\nu\left(\nu+1\right)} \left\{ \left[\Delta\Phi^{(1)}\left(\nu_{i\bullet}+2,\nu+2\right) \right]^{2} \right. \\
&\quad \left. + \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu_{i\bullet}+1\right) \Delta\Phi^{(1)}\left(\nu_{i\bullet}+2,\nu+2\right) + \Delta\Phi^{(2)}\left(\nu_{i\bullet}+2,\nu+2\right) \right\} \\
&\quad - \sum_{i,j} \frac{\nu_{ij}\nu_{i\bullet}}{\nu^{2}} \Delta\Phi^{(1)}\left(\nu_{ij}+1,\nu+1\right) \Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\nu+1\right) ,
\end{aligned} \tag{67}
$$

$$
\begin{aligned}
{}_{OD}\sigma^2_{\overline{IJM}} = \sum_{i,j}\sum_{m\neq i}\frac{\nu_{ij}\nu_{m\bullet}}{\nu(\nu+1)}\left\{\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+2)-\Phi^{(2)}(\nu+2)\right\} \\
-\sum_{i,j}\frac{\nu_{ij}}{\nu}\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+1)\sum_{m\neq i}\frac{\nu_{m\bullet}}{\nu}\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1)\ .
\end{aligned}
\tag{68}
$$

The equation

$$
\Delta\Phi^{(1)}(a,c)=\Delta\Phi^{(1)}(a,b)+\Delta\Phi^{(1)}(b,c) \tag{69}
$$

is useful for accounting for the mixing of variables between the two summations in Equation (67). Again, substitution with appropriate conversions for the $\Delta\Phi^{(1)}$ functions can be used to get consistent variables for cancellation, which produces

$$
\begin{aligned}
{}_{D}\sigma^2_{\overline{IJM}} = \sum_{i,j}\frac{\nu_{ij}(\nu_{i\bullet}+1)}{\nu(\nu+1)}\times \\
\left[\frac{\nu-\nu_{i\bullet}}{(\nu_{i\bullet}+1)(\nu+1)}\left\{\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)+\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+1)+\frac{\nu-\nu_{i\bullet}}{(\nu_{i\bullet}+1)(\nu+1)}\right\}\right. \\
\left.+\Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2)+\frac{(\nu-\nu_{i\bullet})}{(\nu_{i\bullet}+1)\nu}\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\right]
\end{aligned}
\tag{70}
$$

and

$$
\begin{aligned}
{}_{OD}\sigma^2_{\overline{IJM}} = \sum_{i,j}\sum_{m\neq i}\frac{\nu_{ij}\nu_{m\bullet}}{\nu(\nu+1)}\left[-\frac{1}{\nu}\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1)\right. \\
\left.-\frac{1}{\nu+1}\left\{\Delta\Phi^{(1)}(\nu_{ij}+1,\nu+1)+\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1)\right\}+\frac{1}{(\nu+1)^2}-\Phi^{(2)}(\nu+2)\right]\ .
\end{aligned}
\tag{71}
$$

For the next equation,

$$
\sigma^2_{\overline{IM}} = \mathcal{E}_{\overline{IM}} - \mathcal{E}_{\overline{I}}\mathcal{E}_{\overline{M}}\,, \tag{72}
$$

the resulting combination is

$$\sigma^2_{\overline{IM}} = \sum_{i,m\neq i} \frac{\nu_{i\bullet}\nu_{m\bullet}}{\nu(\nu+1)} \left\{ \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+2) - \Phi^{(2)}(\nu+2) \right\}$$
$$+ \sum_{i} \frac{\nu_{i\bullet}(\nu_{i\bullet}+1)}{\nu(\nu+1)} \left\{ \left[\Delta\Phi^{(1)}(\nu_{i\bullet}+2,\nu+2)\right]^2 + \Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2) \right\} \tag{73}$$
$$- \sum_{i} \frac{\nu_{i\bullet}}{\nu} \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1) \sum_{m} \frac{\nu_{m\bullet}}{\nu} \Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1) \ .$$

The same breakdown into diagonal and off-diagonal terms results in

$${}_D\sigma^2_{\overline{IM}} = \sum_{i} \frac{\nu_{i\bullet}(\nu_{i\bullet}+1)}{\nu(\nu+1)} \left\{ \left[\Delta\Phi^{(1)}(\nu_{i\bullet}+2,\nu+2)\right]^2 + \Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2) \right\}$$
$$- \sum_{i} \left(\frac{\nu_{i\bullet}}{\nu} \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1) \right)^2 \tag{74}$$

and

$${}_{OD}\sigma^2_{\overline{IM}} = \sum_{i,m\neq i} \frac{\nu_{i\bullet}\nu_{m\bullet}}{\nu(\nu+1)} \left\{ \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+2)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+2) - \Phi^{(2)}(\nu+2) \right\}$$
$$- \sum_{i,m\neq i} \frac{\nu_{i\bullet}\nu_{m\bullet}}{\nu^2} \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1) \ . \tag{75}$$

Various substitutions and algebraic manipulations produce

$${}_D\sigma^2_{\overline{IM}} = \sum_{i} \frac{\nu_{i\bullet}(\nu_{i\bullet}+1)}{\nu(\nu+1)} \left[\frac{(\nu-\nu_{i\bullet})}{\nu(\nu_{i\bullet}+1)} \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)^2 \right.$$
$$\left. + \frac{\nu-\nu_{i\bullet}}{(\nu_{i\bullet}+1)(\nu+1)} \left(2\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1) + \left(\frac{\nu-\nu_{i\bullet}}{(\nu_{i\bullet}+1)(\nu+1)} \right) \right) + \Delta\Phi^{(2)}(\nu_{i\bullet}+2,\nu+2) \right] \tag{76}$$

and

$${}_{OD}\sigma^2_{\overline{IM}} = \sum_{i,m\neq i} \frac{\nu_{i\bullet}\nu_{m\bullet}}{\nu(\nu+1)} \left[-\frac{1}{\nu} \Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1) \right.$$
$$\left. + \frac{1}{\nu+1} \left[\frac{1}{(\nu+1)} - \left(\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1) + \Delta\Phi^{(1)}(\nu_{m\bullet}+1,\nu+1) \right) \right] - \Phi^{(2)}(\nu+2) \right] . \tag{77}$$

The $\Delta\Phi^{(1)}(x,y)$ function is a common function in the information theoretic formulae. When both the x and y terms are large and close to each other, inaccurate estimates may be obtained if the function is numerically implemented as a difference of digamma function calls using such libraries as can be found in MATLAB. Increased accuracy can be obtained by using the recursion relations of Equations (61) and (62) to estimate the $\Delta\Phi^{(1)}(x,y)$ function for large, similar values.

In working toward a revised algorithm for calculating $\Delta\Phi^{(1)}(x,y)$ with better numerical precision, it should be remembered that Wolpert and Wolf's equations can account for prior probabilities. The result is that the ν, ν_{ij}, $\nu_{i\bullet}$, and $\nu_{\bullet j}$ terms may not always be integers for certain selections of prior probabilities, nor are their differences always integers. This leads to a bit more complexity in the reformulation of $\Delta\Phi^{(1)}(x,y)$ because the recursion relations in Equations (61) and (62) cannot be guaranteed to lead to an advantageous cancellation between the pairs of digamma functions used in estimating $\Delta\Phi^{(1)}$. In general, the differences of the priors may be non-integer and prevent simplification of the $\Delta\Phi^{(1)}$ functions to finite sums by the extraction of integer differences. However, some improvement in numerical accuracy can be achieved by using the recurrence relations to get a function containing a finite sum and the difference of digamma functions for two very similar variables.

The function can be reformulated through the use of the recurrence relationships of Equation (61) to get

$$\begin{aligned}\Delta\Phi^{(1)}(n_1+\rho_1, n_2+\rho_2) &= \Psi_0(n_1+\rho_1) - \Psi_0(n_2+\rho_2) \\ &= \Psi_0(n_1+\rho_1) - \Psi_0(n_1+\rho_1+\mathrm{rem}(\rho_2-\rho_1,1)) - \sum_{r=n_1+\rho_1+\mathrm{rem}(\rho_2-\rho_1,1)}^{n_2+\rho_2-1} \frac{1}{r}\end{aligned} \tag{78}$$

for integer sample counts n_1 and n_2 and prior probability contributions ρ_1 and ρ_2. As described, the variables ρ_1 and ρ_2 may not always be integers.

5.2 DERIVATION OF THE FINAL TERM, $\sigma^2_{\overline{IN}}$

The final sixth term, $\mathfrak{S}_{\overline{IN}}$, is the most difficult one to refactor into an equation for the covariance. It is derived from the first- and second-order statistics to get

$$
\begin{aligned}
\sigma_{\overline{IN}}^2 &= \mathcal{E}_{\overline{IN}} - \mathcal{E}_{\overline{I}}\mathcal{E}_{\overline{N}} \\
&= \sum_{i,n} \frac{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}{\nu\left(\nu+1\right)} \left\{ \left\{ \left[\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right]^2 + \Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right) \right\} \right. \\
&\quad \times \left(1 - \frac{\left(\nu_{i\bullet}-\nu_{in}\right)+\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}} + \frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)} \right) \\
&\quad + \Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right) \times \left[S_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right) + S_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in},\overline{\nu}_{in}\right) \right] \\
&\quad \left. + S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right) \right\} \\
&\quad - \sum_{i,n} \frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu^2} \Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\nu+1\right) \ .
\end{aligned}
\tag{79}
$$

An easy simplification is to use the relation

$$
\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)\left(1 - \frac{\left(\nu_{i\bullet}-\nu_{in}\right)+\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}} + \frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)} \right) = \nu_{i\bullet}\nu_{\bullet n} + \nu_{in} \tag{80}
$$

to get

$$
\begin{aligned}
\sigma_{\overline{IN}}^2 &= \sum_{i,n} \frac{1}{\nu\left(\nu+1\right)} \left\{ \quad \left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right) \left\{ \left[\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right]^2 + \Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right) \right\} \right. \\
&\quad + \overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right) \times \left[S_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right) + S_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in},\overline{\nu}_{in}\right) \right] \\
&\quad \left. + \overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right) S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right) \right\} \\
&\quad - \sum_{i,n} \frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu^2} \Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\nu+1\right) \ .
\end{aligned}
\tag{81}
$$

This also eliminates an awkward pole at $\overline{\nu}_{in} = 0$, which can be considered to be a valid value through a limiting process.

The function S_1 can be refactored for better numerical accuracy. First, replace r with $k = r - 1$ to get

$$
S_1\left(\nu_x,\nu_y,\nu_{in},\overline{\nu}_{in}\right) = -\frac{\left(\nu_x-\nu_{in}\right)\left(\nu_x+1\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)} + \sum_{k=1}^{\infty} \frac{1}{k\left(k+1\right)} \left[\frac{\left(\nu_x-\nu_{in}\right)_{k+1}}{\left(\overline{\nu}_{in}\right)_{k+1}} \left(1 - \frac{\nu_y-\nu_{in}}{\overline{\nu}_{in}+k+1} \right) \right] \ . \tag{82}
$$

Use the equality,

$$
\left(\overline{\nu}_{in}+k+1\right)\left(\overline{\nu}_{in}\right)_{k+1} = \overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)\left(\overline{\nu}_{in}+2\right)_k , \tag{83}
$$

to get

$$S_1\left(\nu_x,\nu_y,\nu_{in},\overline{\nu}_{in}\right)=-\frac{\left(\nu_x-\nu_{in}\right)\left(\nu_x+1\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}+\frac{1}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[\frac{\left(\nu_x-\nu_{in}\right)_{k+1}}{\left(\overline{\nu}_{in}+2\right)_k}\left(\nu_x+k+1\right)\right] \tag{84}$$

Using similar steps as used to get the equality in Equation (83), the equation

$$\left(\nu_x-\nu_{in}\right)_{k+1}=\left(\nu_x-\nu_{in}\right)\left(\nu_x-\nu_{in}+1\right)_k \tag{85}$$

can be used to reform S_1 to

$$\begin{aligned}S_1\left(\nu_x,\nu_y,\nu_{in},\overline{\nu}_{in}\right)=&-\frac{\left(\nu_x-\nu_{in}\right)\left(\nu_x+1\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}+\\&\frac{\left(\nu_x-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[\frac{\left(\nu_x-\nu_{in}+1\right)_k}{\left(\overline{\nu}_{in}+2\right)_k}\left(\nu_x+k+1\right)\right].\end{aligned} \tag{86}$$

Better numerical accuracy can be obtained for S_1 by the use of judicious summation. The first term in Equation (86) contains factors that can be extracted from the summation,

$$S_1\left(\nu_x,\nu_y,\nu_{in},\overline{\nu}_{in}\right)=-\frac{\left(\nu_x-\nu_{in}\right)\left(\nu_x+1\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}\left(1-\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[\frac{\left(\nu_x-\nu_{in}+1\right)_k}{\left(\overline{\nu}_{in}+2\right)_k}\frac{\left(\nu_x+k+1\right)}{\left(\nu_x+1\right)}\right]\right). \tag{87}$$

To continue the derivation, define a function

$$\mathrm{F}_1\left(x,y,z\right)=1-\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\frac{\left(x+k+1\right)\left(x-z+1\right)_k}{\left(x+1\right)\left(x+y-z+2\right)_k}\ . \tag{88}$$

Better numerical accuracy can be achieved by the cancellation of multiple small values instead of two larger values. The function

$$\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}=1 \tag{89}$$

can be used to redefine $\mathrm{F}_1\left(x,y,z\right)$ as

$$\mathrm{F}_1\left(x,y,z\right)=\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[1-\frac{\left(x+k+1\right)\left(x-z+1\right)_k}{\left(x+1\right)\left(x+y-z+2\right)_k}\right]\ . \tag{90}$$

Now

$$S_1\left(\nu_x,\nu_y,\nu_{in},\overline{\nu}_{in}\right)=-\frac{\left(\nu_x-\nu_{in}\right)\left(\nu_x+1\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}F_1\left(\nu_x,\nu_y,\nu_{in}\right)\ . \tag{91}$$

For the numerical summation of $\mathrm{F}_1\left(x,y,z\right)$, the second term inside the square brackets eventually drives toward zero. Once this value is less than the machine precision of the numerical calculation, it can be neglected and the rest of the summation analytical performed with the function

$$\sum_{k=q}^{\infty}\frac{1}{k\left(k+1\right)}=\frac{1}{q}\ . \tag{92}$$

The double summation

$$\begin{aligned}S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right)=&\frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\left(\overline{\nu}_{in}\right)_2}\\&-\left(\nu_{i\bullet}-\nu_{in}\right)\sum_{s=2}^{\infty}\frac{\left(\nu_{\bullet n}-\nu_{in}\right)_s}{\left(\overline{\nu}_{in}\right)_{1+s}}\frac{1}{s\left(s-1\right)}\\&-\left(\nu_{\bullet n}-\nu_{in}\right)\sum_{r=2}^{\infty}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r}{\left(\overline{\nu}_{in}\right)_{r+1}}\frac{1}{r\left(r-1\right)}\\&+\sum_{r=2}^{\infty}\sum_{s=2}^{\infty}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_r\left(\nu_{\bullet n}-\nu_{in}\right)_s}{\left(\overline{\nu}_{in}\right)_{r+s}}\frac{1}{r\left(r-1\right)}\frac{1}{s\left(s-1\right)}\end{aligned} \tag{93}$$

can be rewritten by switching from r and s to k and l, where $k=r-1$ and $l=s-1$:

$$\begin{aligned}S_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in},\overline{\nu}_{in}\right)=&\frac{\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)}{\overline{\nu}_{in}\left(\overline{\nu}_{in}+1\right)}\\&-\left(\nu_{i\bullet}-\nu_{in}\right)\sum_{l=1}^{\infty}\frac{1}{l\left(l+1\right)}\frac{\left(\nu_{\bullet n}-\nu_{in}\right)_{l+1}}{\left(\overline{\nu}_{in}\right)_{l+2}}\\&-\left(\nu_{\bullet n}-\nu_{in}\right)\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_{k+1}}{\left(\overline{\nu}_{in}\right)_{k+2}}\\&+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{1}{k\left(k+1\right)}\frac{1}{l\left(l+1\right)}\frac{\left(\nu_{i\bullet}-\nu_{in}\right)_{k+1}\left(\nu_{\bullet n}-\nu_{in}\right)_{l+1}}{\left(\overline{\nu}_{in}\right)_{k+l+2}}\ .\end{aligned} \tag{94}$$

Next, terms from the sums are extracted to get

$$\begin{aligned} S_2 = & \frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{\overline{\nu}_{in}(\overline{\nu}_{in}+1)} \\ & -(\nu_{i\bullet}-\nu_{in})\frac{(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})_2}\sum_{l=1}^{\infty}\frac{1}{l(l+1)}\frac{(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_l} \\ & -(\nu_{\bullet n}-\nu_{in})\frac{(\nu_{i\bullet}-\nu_{in})}{(\overline{\nu}_{in})_2}\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k}{(\overline{\nu}_{in}+2)_k} \\ & +\frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})_2}\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{1}{k(k+1)}\frac{1}{l(l+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_{k+l}} \end{aligned} \tag{95}$$

using

$$(\nu_{\bullet n}-\nu_{in})_{l+1} = (\nu_{\bullet n}-\nu_{in})(\nu_{\bullet n}-\nu_{in}+1)_l \tag{96}$$

and

$$(\overline{\nu}_{in})_2 = (\overline{\nu}_{in})(\overline{\nu}_{in}+1)\ . \tag{97}$$

Common terms are again extracted to get

$$\begin{aligned} S_2 = & \frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)}\Bigg[1 \\ & -\sum_{l=1}^{\infty}\frac{1}{l(l+1)}\frac{(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_l} \\ & -\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k}{(\overline{\nu}_{in}+2)_k} \\ & +\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{1}{k(k+1)}\frac{1}{l(l+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_{k+l}}\Bigg]\ . \end{aligned} \tag{98}$$

The terms in the equation can be rearranged to give

$$\begin{aligned} S_2 = & \frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)}\Bigg[\Bigg\{1-\sum_{l=1}^{\infty}\frac{1}{l(l+1)}\frac{(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_l}\Bigg\} \\ & +\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{1}{k(k+1)}\frac{1}{l(l+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k(\nu_{\bullet n}-\nu_{in}+1)_l}{(\overline{\nu}_{in}+2)_{k+l}}-\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k}{(\overline{\nu}_{in}+2)_k}\Bigg]\ . \end{aligned} \tag{99}$$

The expansion

$$(a)_{k+l} = \frac{\Gamma(a+k+l)}{\Gamma(a)} = \frac{\Gamma(a+k)}{\Gamma(a)}\frac{\Gamma(a+k+l)}{\Gamma(a+k)} = (a)_k (a+k)_l \tag{100}$$

can be used to get

$$\begin{aligned} S_2 = \frac{(\nu_{i\bullet} - \nu_{in})(\nu_{\bullet n} - \nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)} \Bigg[& \left\{ 1 - \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \frac{(\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2)_l} \right\} \\ & + \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \frac{1}{k(k+1)} \frac{1}{l(l+1)} \frac{(\nu_{i\bullet} - \nu_{in} + 1)_k (\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2)_k (\overline{\nu}_{in}+2+k)_l} - \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \frac{(\nu_{i\bullet} - \nu_{in} + 1)_k}{(\overline{\nu}_{in}+2)_k} \Bigg] \end{aligned} \tag{101}$$

and terms rearranged to get

$$\begin{aligned} S_2 = \frac{(\nu_{i\bullet} - \nu_{in})(\nu_{\bullet n} - \nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)} \Bigg[& \left\{ 1 - \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \frac{(\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2)_l} \right\} \\ & - \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \frac{(\nu_{i\bullet} - \nu_{in} + 1)_k}{(\overline{\nu}_{in}+2)_k} \left\{ 1 - \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \frac{(\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2+k)_l} \right\} \Bigg] . \end{aligned} \tag{102}$$

Using a similar approach to that used in Equations (88) through (90), Equation (102) can be rearranged as

$$\begin{aligned} S_2 = \frac{(\nu_{i\bullet} - \nu_{in})(\nu_{\bullet n} - \nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)} \Bigg[& \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \left\{ 1 - \frac{(\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2)_l} \right\} \\ & - \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \frac{(\nu_{i\bullet} - \nu_{in} + 1)_k}{(\overline{\nu}_{in}+2)_k} \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \left\{ 1 - \frac{(\nu_{\bullet n} - \nu_{in} + 1)_l}{(\overline{\nu}_{in}+2+k)_l} \right\} \Bigg] . \end{aligned} \tag{103}$$

Next, the following functions are defined:

$$\mathrm{F}_2(x,y,z) = 1 - \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \frac{(y-z+1)_k}{(x+y-z+2)_k} \tag{104}$$

and

$$\mathrm{F}_3(x,y,z) = \sum_{k=1}^{\infty} \frac{(x-z+1)_k}{k(k+1)(x+y-z+2)_k} \sum_{l=1}^{\infty} \frac{1}{l(l+1)} \left\{ 1 - \frac{(y-z+1)_l}{(x+y-z+2+k)_l} \right\} , \tag{105}$$

or equivalently

$$\mathrm{F}_3(x,y,z)=\sum_{k=1}^{\infty}\frac{(x-z+1)_k}{k(k+1)(x+y-z+2)_k}\mathrm{F}_2(x+k,y,z)\ . \tag{106}$$

Then

$$S_2=\frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)}\left[\mathrm{F}_2(\nu_{\bullet n},\nu_{i\bullet},\nu_{in})-\mathrm{F}_3(\nu_{\bullet n},\nu_{i\bullet},\nu_{in})\right], \tag{107}$$

or, by symmetry,

$$S_2=\frac{(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})}{(\overline{\nu}_{in})(\overline{\nu}_{in}+1)}\left[\mathrm{F}_2(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})-\mathrm{F}_3(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})\right]\ . \tag{108}$$

The functions $\mathrm{F}_2(\nu_x,\nu_y,\nu_{in})$ and $\mathrm{F}_3(\nu_x,\nu_y,\nu_{in})$ can be implemented in computer code for improved numerical accuracy. The choice between using Equation (107) or (108) can have a significant impact on the numerical accuracy of the results when calculated with finite precision computers. More accurate results are achieved when the first term is the larger of $\nu_{i\bullet}$ and $\nu_{\bullet n}$ in functions $\mathrm{F}_2(\nu_x,\nu_y,\nu_{in})$ and $\mathrm{F}_3(\nu_x,\nu_y,\nu_{in})$.

With the reformulated functions, the equation for $\sigma^2_{\overline{IN}}$ updates to

$$\begin{aligned}\sigma^2_{\overline{IN}}=\sum_{i,n}\frac{1}{\nu(\nu+1)}\Big\{\ &(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in})\left\{\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)^2+\Delta\Phi^{(2)}(\overline{\nu}_{in}+2,\nu+2)\right\}\\ &-\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)\left\{(\nu_{\bullet n}-\nu_{in})(\nu_{\bullet n}+1)\mathrm{F}_1(\nu_{\bullet n},\nu_{i\bullet},\nu_{in})+(\nu_{i\bullet}-\nu_{in})(\nu_{i\bullet}+1)\mathrm{F}_1(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})\right\}\\ &+(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})\left[\mathrm{F}_2(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})-\mathrm{F}_3(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})\right]\Big\}\\ &-\frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu^2}\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{\bullet n}+1,\nu+1)\ .\end{aligned} \tag{109}$$

Next, the equation can be reorganized to move related terms closer together:

$$\begin{aligned}\sigma_{\overline{IN}}^2 = \sum_{i,n}\frac{1}{\nu(\nu+1)}\Big\{ & (\nu_{i\bullet}\nu_{\bullet n}+\nu_{in})\Delta\Phi^{(2)}(\overline{\nu}_{in}+2,\nu+2) \\ & +(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in})\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)^2 - \frac{\nu_{i\bullet}\nu_{\bullet n}(\nu+1)}{\nu}\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{\bullet n}+1,\nu+1) \\ & -\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)\{(\nu_{\bullet n}-\nu_{in})(\nu_{\bullet n}+1)\mathrm{F}_1(\nu_{\bullet n},\nu_{i\bullet},\nu_{in})+(\nu_{i\bullet}-\nu_{in})(\nu_{i\bullet}+1)\mathrm{F}_1(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})\} \\ & +(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})[\mathrm{F}_2(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})-\mathrm{F}_3(\nu_{i\bullet},\nu_{\bullet n},\nu_{in})]\Big\}\ .\end{aligned} \tag{110}$$

The relationships

$$\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)=\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\overline{\nu}_{in}+1)+\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1) \tag{111}$$

and

$$\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)=\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1)+\frac{1}{\overline{\nu}_{in}+1}-\frac{1}{\nu+1}, \tag{112}$$

or equivalently

$$\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)=\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1)+\frac{(\nu-\overline{\nu}_{in})}{(\overline{\nu}_{in}+1)(\nu+1)}, \tag{113}$$

can be used to further simplify $\sigma_{\overline{IN}}^2$.

A new variable is defined to continue to refine the equation for $\sigma_{\overline{IN}}^2$ through a number of steps:

$$T_1=(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in})\Delta\Phi^{(1)}(\overline{\nu}_{in}+2,\nu+2)^2-\frac{\nu_{i\bullet}\nu_{\bullet n}(\nu+1)}{\nu}\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\nu+1)\Delta\Phi^{(1)}(\nu_{\bullet n}+1,\nu+1)\ . \tag{114}$$

Terms are rearranged to get

$$\begin{aligned}T_1 = & (\nu_{i\bullet}\nu_{\bullet n}+\nu_{in})\left(\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1)+\frac{(\nu-\overline{\nu}_{in})}{(\overline{\nu}_{in}+1)(\nu+1)}\right)^2 \\ & -\frac{\nu_{i\bullet}\nu_{\bullet n}(\nu+1)}{\nu}\left(\Delta\Phi^{(1)}(\nu_{i\bullet}+1,\overline{\nu}_{in}+1)+\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1)\right)\times \\ & \left(\Delta\Phi^{(1)}(\nu_{\bullet n}+1,\overline{\nu}_{in}+1)+\Delta\Phi^{(1)}(\overline{\nu}_{in}+1,\nu+1)\right)\ .\end{aligned} \tag{115}$$

Reformulation continues to

$$
\begin{aligned}
T_1 = &\left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right)\times \\
&\left(\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2 + 2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) + \left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
&-\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\left(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\right)\times \\
&\left(\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\right) .
\end{aligned}
\tag{116}
$$

The variable can be reduced to

$$
\begin{aligned}
T_1 = &\left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right)\times \\
&\left(\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2 + 2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) + \left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
&-\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\left(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right. \\
&+\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) \\
&+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right) \\
&\left.+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\right) .
\end{aligned}
\tag{117}
$$

Squared terms are collected to get

$$
\begin{aligned}
T_1 = &\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\left[\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)-\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\right] \\
&+\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left(2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)+\left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
&-\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\left(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right. \\
&+\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) \\
&\left.+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right)
\end{aligned}
\tag{118}
$$

and reduced to

$$
\begin{aligned}
T_1 = {} & \Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\left[\nu_{in}-\frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu}\right] \\
& +\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left(2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)+\left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
& -\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\left(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right. \\
& \qquad +\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) \\
& \qquad \left.+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right) .
\end{aligned}
\tag{119}
$$

Fold the result back into the equation for $\sigma^2_{\overline{IN}}$ to get

$$
\begin{aligned}
\sigma^2_{\overline{IN}} = {} & \sum_{i,n}\frac{1}{\nu\left(\nu+1\right)}\left\{\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right)\right. \\
& +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\left[\nu_{in}-\frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu}\right] \\
& +\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left(2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)+\left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
& -\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\left(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right. \\
& \qquad +\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) \\
& \qquad \left.+\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\right) \\
& -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in}\right) \\
& -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right) \\
& \left.+\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)\left[\mathrm{F}_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)-\mathrm{F}_3\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)\right]\right\} .
\end{aligned}
\tag{120}
$$

The remaining $\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+2,\nu+2\right)$ term is expanded to get

$$
\begin{aligned}
\sigma^2_{\overline{IN}} = \sum_{i,n}\frac{1}{\nu(\nu+1)}\Big\{ & \left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right) \\
& +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\left[\nu_{in}-\frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu}\right] \\
& +\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left(2\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)+\left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2\right) \\
& -\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Big(\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right) \\
& \qquad +\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right) \\
& \qquad +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\Big) \\
& -\left(\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)+\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right)\times \\
& \left\{\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in}\right)+\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)\right\} \\
& +\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)\left[\mathrm{F}_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)-\mathrm{F}_3\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)\right]\Big\} \ .
\end{aligned}
\tag{121}
$$

Common terms in $\Delta\Phi^{(1)}$ are collected:

$$
\begin{aligned}
\sigma^2_{\overline{IN}} = \frac{1}{\nu(\nu+1)}\sum_{i,n}\Big\{ & \left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2,\nu+2\right)+\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left[\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right]^2 \\
& +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)^2\left[\nu_{in}-\frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu}\right] \\
& +\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)\left(\begin{array}{l}
\dfrac{2\left(\nu_{i\bullet}\nu_{\bullet n}+\nu_{in}\right)\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)} \\
-\dfrac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right) \\
-\dfrac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right) \\
-\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in}\right) \\
-\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)
\end{array}\right) \\
& -\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right) \\
& -\frac{\left(\nu-\overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\Big\{\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in}\right)+\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right) \\
& +\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)\left[\mathrm{F}_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)-\mathrm{F}_3\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)\right]\Big\} \ .
\end{aligned}
\tag{122}
$$

This provides a reformulation of $\sigma^2_{\overline{IN}}$ with better numerical stability.

This page intentionally left blank.

6. NUMERICAL RESULTS

The same process used to evaluate the original MATLAB code was used to evaluate the first revision of the code, based upon the derivations of Section 5. Figure 3 shows a plot of the absolute values of the eigenvalues for the original MATLAB code and the revised code for different multiplier scales applied to the accumulation matrix in Equation (54). As the scale factor increases, the eigenvalues decrease, corresponding to more accurate estimates caused by the increased sample size. Negative eigenvalues are shown in red and indicate where the algorithm has failed to accurately calculate the components of the covariance matrix. The original algorithm fails to estimate good variance and covariance values for sample sizes on the order of 10^7. The revised version is able to handle sample sizes on the order of 10^{11} before the results become corrupted. Although positive, the smallest eigenvalue for the revised code at sample sizes of 10^{12} is suspect because it departs from the relatively linear trend in the log-log plot. Clearly, the revised code fails at 10^{13} because a negative covariance eigenvalue is estimated for the scaled test matrix.

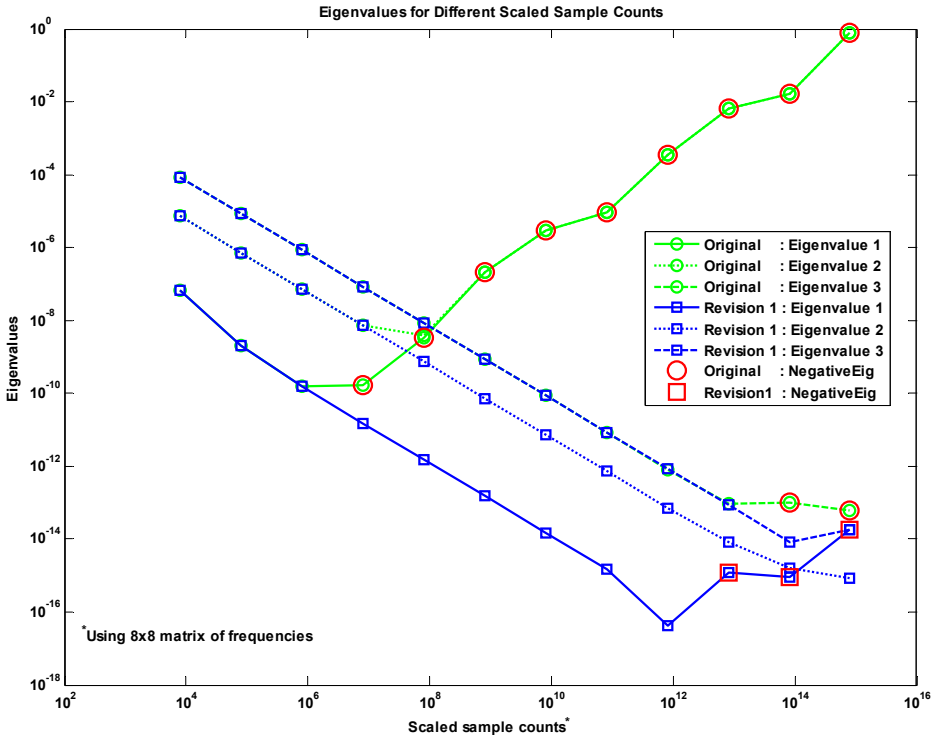


Figure 3. A plot of the absolute value of the eigenvalues of covariance matrices calculated for an accumulation matrix with different scale factors for two different software formulations of the information theoretic equations. The x-axis is the scaled sample counts in the matrix. Negative eigenvalues are shown with red circles and squares to indicate that the covariance terms are incorrectly estimated.

Figure 4 shows the computation time for the original MATLAB code and the revised code for different scales of the accumulation matrix in Equation (54). Computation time is significantly reduced for most values in the matrix with the revised code, except for a range of values associated with the scale factor range of 10^9 to 10^{11}. This is caused by the execution of the functions associated with the infinite sums. The maximum difference is less than a factor of 10; the increase in computation time could have been many orders of magnitude higher if the refactoring had led to an inefficient algorithm. Although the execution time for reformulation of the computer code was secondary to achieving improved numerical precision, the execution time was tracked so that changes in computation time for different formulations could be monitored.

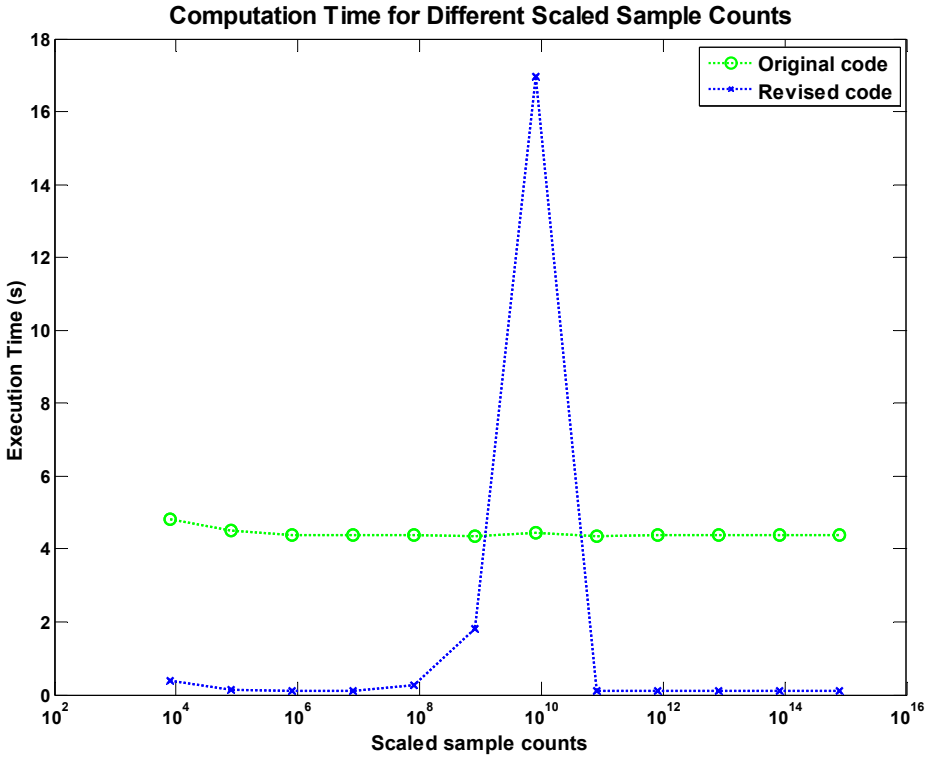


Figure 4. The computation time for calculating the entropies and covariance matrices for different scaling of an example accumulation matrix with the original MATLAB code and the revised MATLAB code.

It is recognized that the presented results are for only one 8×8 accumulation matrix. Other accumulation matrices may fail at lower sample counts. Specifically, matrices with large counts in a few bins, such as is the case for the evaluation of tracking algorithms, may be susceptible to inaccurate estimates for lower sample sizes. Despite continued numerical instability of the revised algorithm, the

optimization of the algorithm has extended the accuracy of the evaluation to sample sizes five orders of magnitude beyond the original implementation.

The primary variable suffering the most from numerical stability issues is $\sigma^2_{\overline{IN}}$. The four variables, ν, ν_{in}, $\nu_{i\bullet}$, and $\nu_{\bullet n}$ can be selected as the fundamental variables in the equations for $\sigma^2_{\overline{IN}}$. All other variables can be defined as combinations of these four. As previously discussed, the accumulation matrices used to evaluate tracking algorithms have very large counts in the true negatives bin. Generally, there are three classes of number ranges for which the evaluation has to estimate terms: the class where ν_{in} is the true negatives bin: {large ν_{in}, $\nu_{i\bullet}$, and $\nu_{\bullet n}$}; the class where either the row or column includes the true negatives bin {small ν_{in}, small $\nu_{i\bullet}$, large $\nu_{\bullet n}$} or {small ν_{in}, large $\nu_{i\bullet}$, small $\nu_{\bullet n}$}; and the class of counts of the number of times that pairs of tracks from two data sets associate: {small ν_{in}, small $\nu_{i\bullet}$, small $\nu_{\bullet n}$}. The first two classes encounter numerical stability problems long before the third class. Thus, only the first two classes have to be examined for potential techniques to improve numerical stability. The numerical stability of the third class will improve by default.

To assess the numerical accuracy of the improved equations, the first two sets were examined for numerical stability issues. Different combinations of ν, ν_{in}, $\nu_{i\bullet}$, and $\nu_{\bullet n}$ were selected and scaled by different orders of magnitude, ranging from 1 to 10^{12}. A few samples are plotted in Figures 5 and 6. For Figure 5, the case where all variables are large values, the calculation begins to lose precision at a scale factor of 10^8, or when the counts are around 10^{14}. For Figure 6, where one variable is large and the other two small, the calculation begins to lose precision at a scale factor of 10^9, or when the counts are around 10^{15}.

The numerical stability of this first revision is acceptable for most applications, but because the accumulation matrices for trackers can have extremely large values for the count of true negatives, numerical stability would be advantageous for count sizes of 10^{16} for $\nu_{i\bullet}$, and $\nu_{\bullet n}$. The next section continues to restructure the equation for $\sigma^2_{\overline{IN}}$ to get additional improvement in the numerical stability.

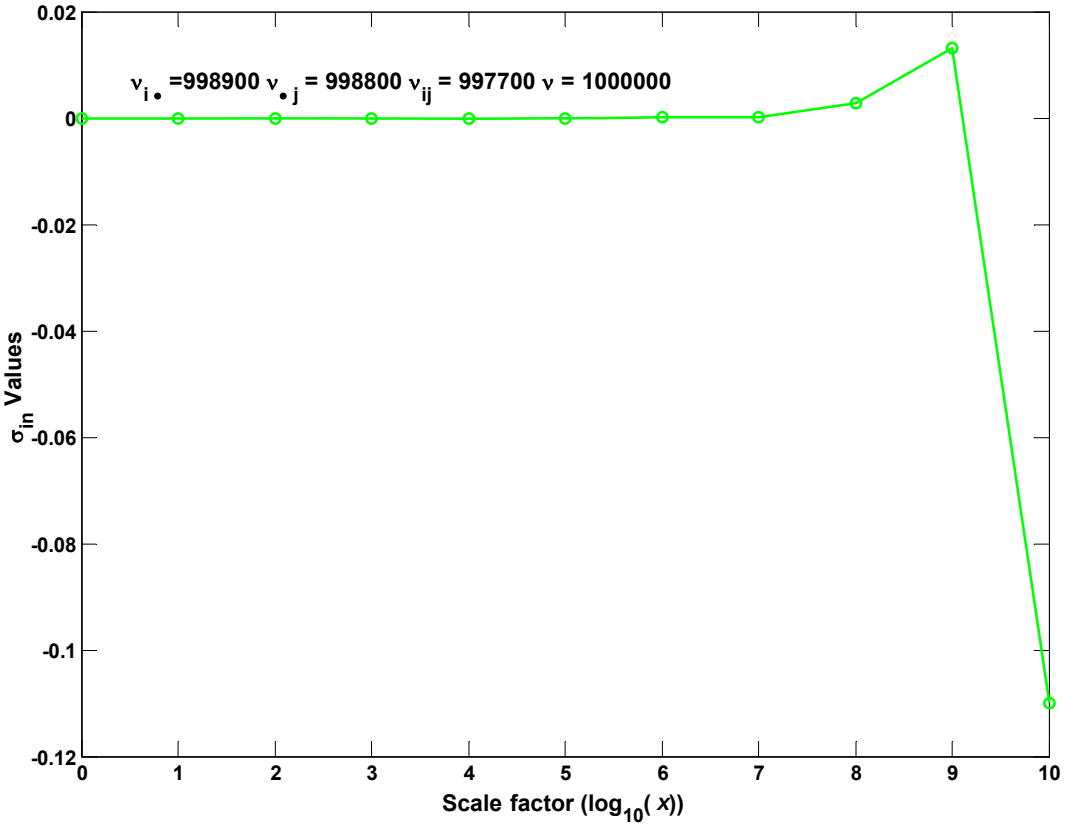


Figure 5. A sample plot of the first class of count distributions for $\sigma_{\overline{IN}}$ *, where all counts are large.*

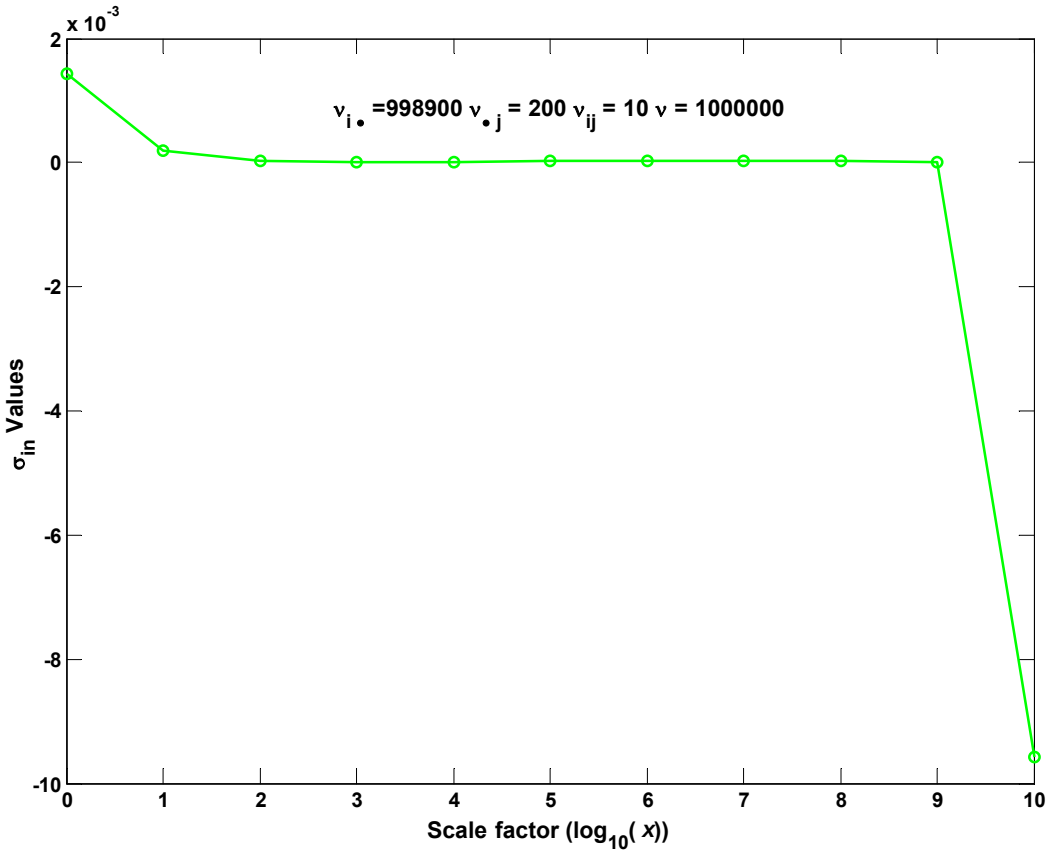


Figure 6. A sample plot of the second class of count distributions for $\sigma_{\overline{IN}}$*, where the row or column count sum is large and the other counts are small.*

This page intentionally left blank.

7. FURTHER REFINEMENT OF THE NUMERICAL ESTIMATES

To further refine the organization of $\sigma^2_{\overline{IN}}$ to get better numerical stability for tracking algorithms, the terms in $\sigma^2_{\overline{IN}}$ were examined to identify possible combinations of sub-terms that would cancel as the scale of the numbers $\nu, \nu_{in}, \nu_{i\bullet}$, and $\nu_{\bullet n}$ increased. The sub-terms selected from $\sigma^2_{\overline{IN}}$ are defined as:

$$\alpha = \left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right)\left(\frac{\left(\nu - \overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\right)^2, \tag{123}$$

$$\beta = \left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right)\Delta\Phi^{(2)}\left(\overline{\nu}_{in}+2, \nu+2\right), \tag{124}$$

$$\gamma = \Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)^2\left[\nu_{in} - \frac{\nu_{i\bullet}\nu_{\bullet n}}{\nu}\right], \tag{125}$$

$$\delta_1 = \Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)\frac{2\left(\nu_{i\bullet}\nu_{\bullet n} + \nu_{in}\right)\left(\nu - \overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}, \tag{126}$$

$$\delta_2 = -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1, \overline{\nu}_{in}+1\right), \tag{127}$$

$$\delta_3 = -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1, \overline{\nu}_{in}+1\right), \tag{128}$$

$$\delta_4 = -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)\left(\nu_{\bullet n} - \nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n}, \nu_{i\bullet}, \nu_{in}\right), \tag{129}$$

$$\delta_5 = -\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1, \nu+1\right)\left(\nu_{i\bullet} - \nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}\right), \tag{130}$$

$$\varepsilon = -\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1, \overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1, \overline{\nu}_{in}+1\right), \tag{131}$$

$$\varsigma_1 = -\frac{\left(\nu - \overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\left(\nu_{\bullet n} - \nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n}, \nu_{i\bullet}, \nu_{in}\right), \tag{132}$$

$$\varsigma_2 = -\frac{\left(\nu - \overline{\nu}_{in}\right)}{\left(\overline{\nu}_{in}+1\right)\left(\nu+1\right)}\left(\nu_{i\bullet} - \nu_{in}\right)\left(\nu_{i\bullet}+1\right)\mathrm{F}_1\left(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}\right), \tag{133}$$

$$\eta_1 = \left(\nu_{i\bullet} - \nu_{in}\right)\left(\nu_{\bullet n} - \nu_{in}\right)\mathrm{F}_2\left(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}\right), \text{ and} \tag{134}$$

$$\eta_2 = -\left(\nu_{i\bullet} - \nu_{in}\right)\left(\nu_{\bullet n} - \nu_{in}\right)\mathrm{F}_3\left(\nu_{i\bullet}, \nu_{\bullet n}, \nu_{in}\right). \tag{135}$$

Figure 7 shows how the sub-terms α, β, γ, δ_1, and ς change as the scale is increased for one selection of ν, ν_{in}, $\nu_{i\bullet}$, and $\nu_{\bullet n}$. The plotted data are divided by $\left(\nu+1\right)$ because these terms then converge to a constant value as the scale factor increases.

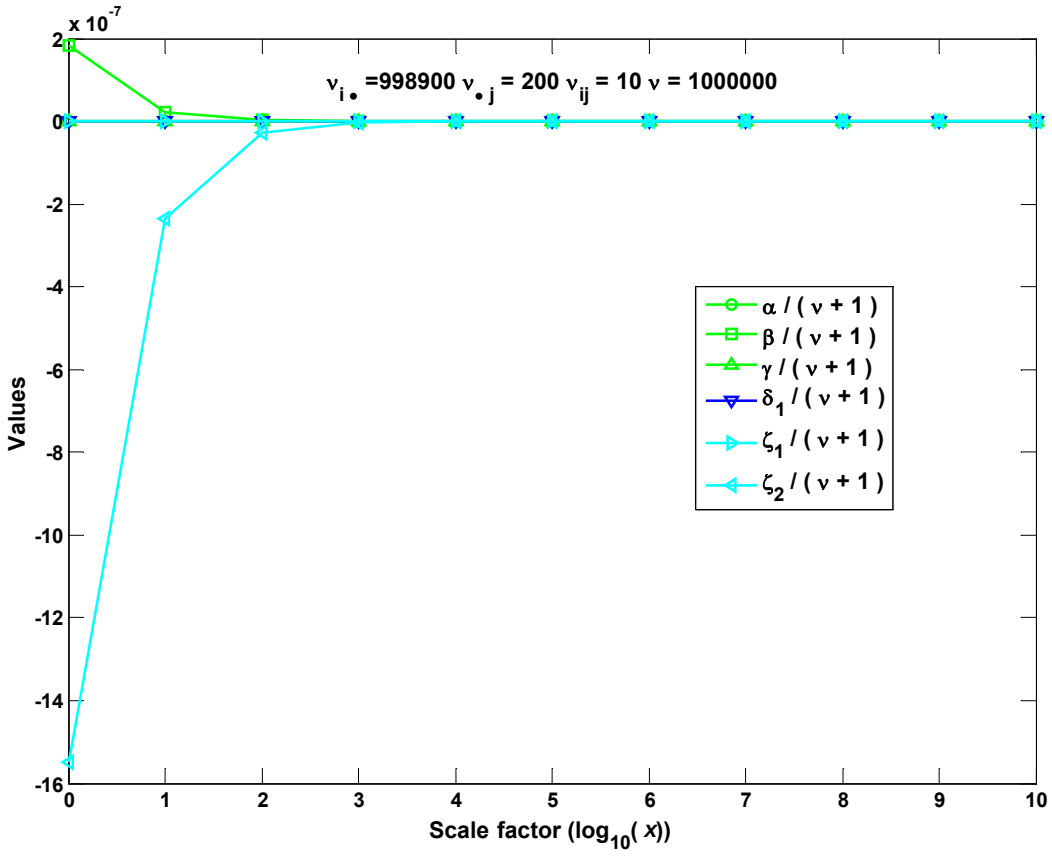


Figure 7. The sub-term values of $\sigma^2_{\overline{IN}}$ *that converge as a function of increasing scale factor.*

Unfortunately, it can be seen in Figures 8 and 9 that the terms δ_2 through δ_5, ε, η_1 and η_2 do not independently converge when divided by $(\nu+1)$. Fortunately, Figure 10 shows that the combinations $\delta_2+\delta_4$, $\delta_3+\delta_5$, and $\varepsilon+\eta_1+\eta_2$ nearly cancel as the scale factor increases until round-off problems creep in above 10^9. The pairs can be reformulated as combined single and double summations to get some additional improvement in numerical stability.

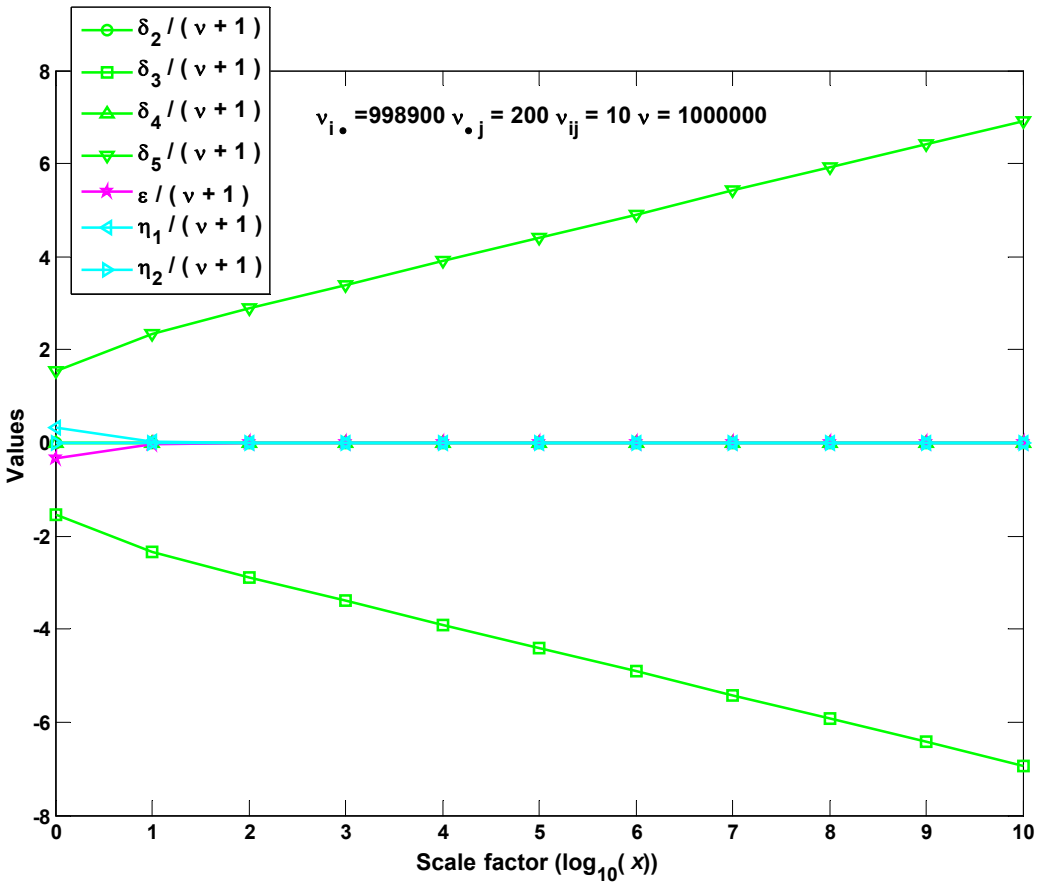


Figure 8. The sub-terms of σ^2_{IN} *that do not converge to limiting values as a function of increasing scale factor.*

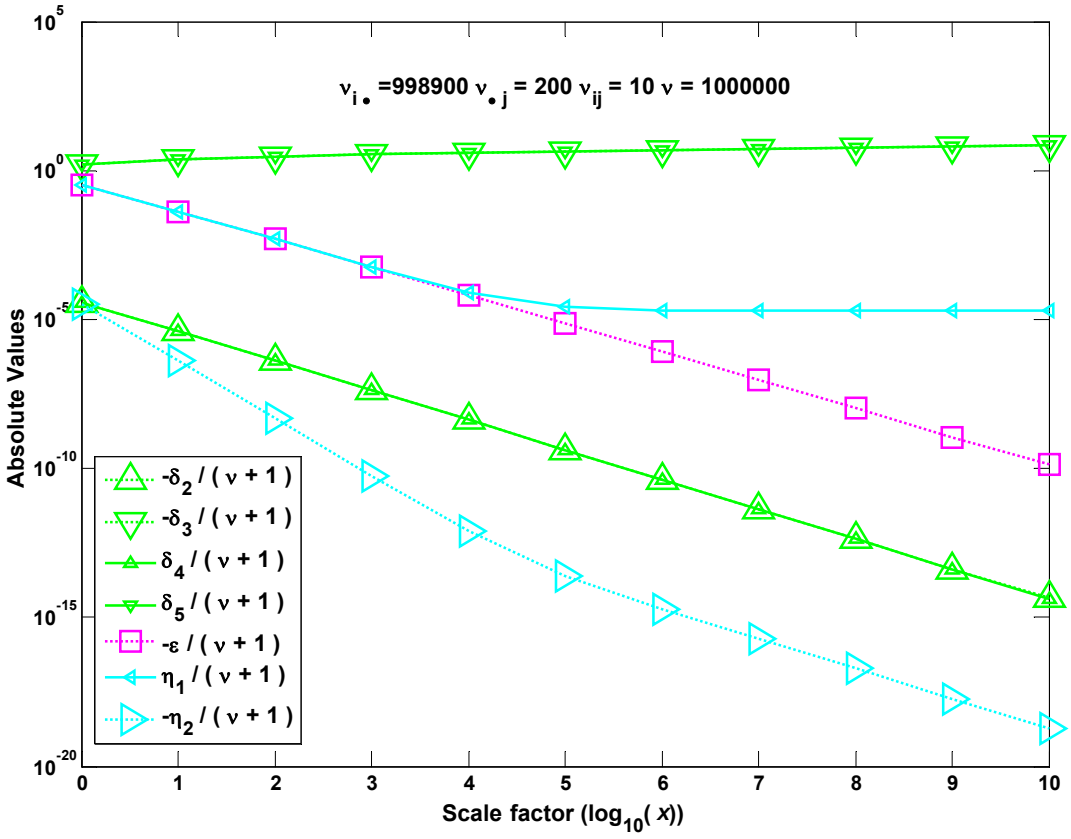


Figure 9. Non-converging sub-terms of $\sigma^2_{\overline{IN}}$ *plotted as logs of absolute values.*

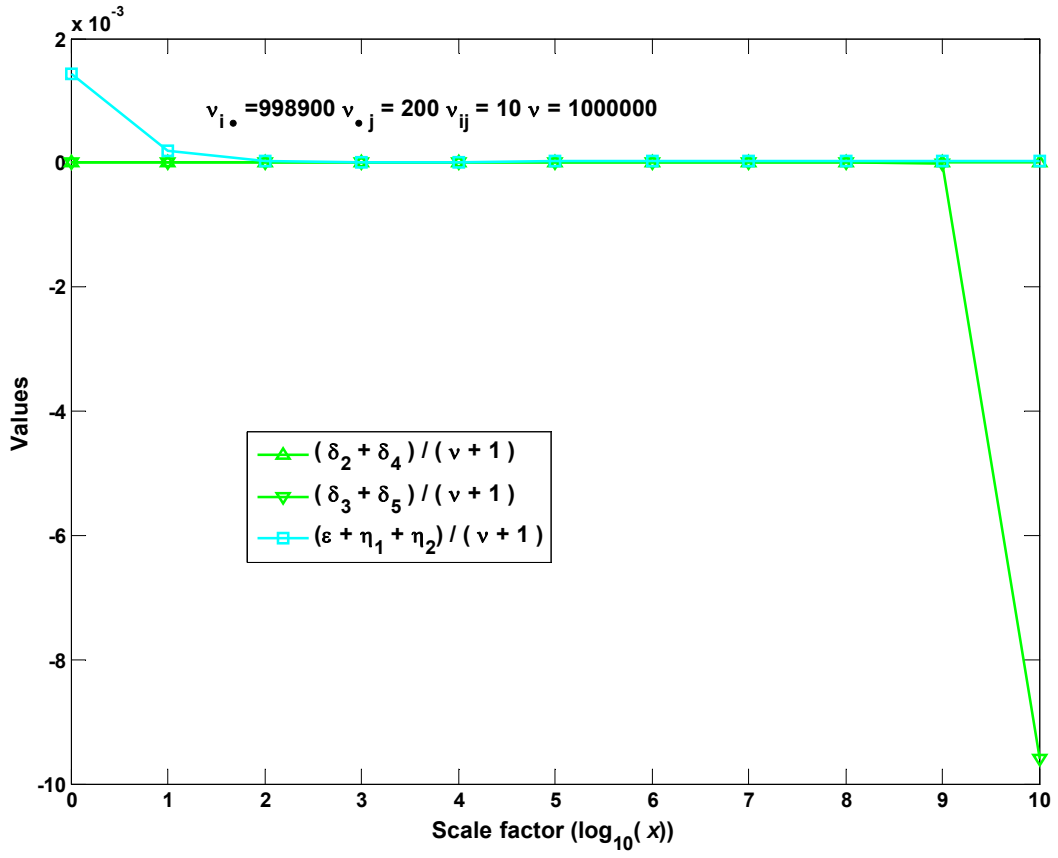


Figure 10. Selected sums of sub-terms of σ_{IN}^2 *with significant cancellation.*

The reformulation of the combination of sums to get improved cancellation requires a summation for $\Delta\Phi^{(1)}$ that is in a form similar to the summations of F_1, F_2, and F_3. These reformulations improved the numerical accuracy by folding constant terms into the summation. The difference of the digamma functions $\Delta\Phi^{(1)}$ can be defined as an infinite sum or as a hypergeometric function of unity argument (z = 1) [13]:

$$\Psi_0(\nu)-\Psi_0(\mu)=\sum_{n=0}^{\infty}\frac{(\nu-\mu)_{n+1}}{(n+1)(\nu)_{n+1}}=\sum_{k=1}^{\infty}\frac{(\nu-\mu)_k}{k(\nu)_k} \quad . \tag{136}$$

Ignoring the common factor of $-\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)$ in the pairs of deltas, the sum for the unique terms in $\delta_2+\delta_4$ is

$$S\left(\delta_2+\delta_4\right)=\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)+\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\mathrm{F}_1\left(\nu_{\bullet n},\nu_{i\bullet},\nu_{in}\right)\ , \tag{137}$$

which is the pair of summations,

$$\begin{aligned}S\left(\delta_2+\delta_4\right)=&\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\sum_{k=1}^{\infty}\frac{\left(\nu_{\bullet n}-\nu_{in}\right)_k}{k\left(\overline{\nu}_{in}+1\right)_k}\\&-\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[1-\frac{\left(\nu_{\bullet n}+k+1\right)\left(\nu_{\bullet n}-\nu_{in}+1\right)_k}{\left(\nu_{\bullet n}+1\right)\left(\overline{\nu}_{in}+2\right)_k}\right]\ .\end{aligned} \tag{138}$$

After combining sums and rearranging terms,

$$\begin{aligned}S\left(\delta_2+\delta_4\right)=&-\left(\nu_{\bullet n}-\nu_{in}\right)\left(\nu_{\bullet n}+1\right)\times\\&\left\{\sum_{k=1}^{\infty}\frac{1}{k\left(k+1\right)}\left[1-\frac{\left(\nu_{\bullet n}-\nu_{in}+1\right)_k}{\left(\overline{\nu}_{in}+1\right)_k}\left(\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\left(\nu_{\bullet n}+1\right)\nu}\frac{\left(k+1\right)}{\left(\nu_{\bullet n}-\nu_{in}+k\right)}+\frac{\left(\overline{\nu}_{in}+1\right)\left(\nu_{\bullet n}+k+1\right)}{\left(\overline{\nu}_{in}+1+k\right)\left(\nu_{\bullet n}+1\right)}\right)\right]\right\}\ .\end{aligned} \tag{139}$$

The identity

$$\left(a+1\right)\left(a+2\right)_k=\left(a+1+k\right)\left(a+1\right)_k \tag{140}$$

has been used to obtain common Pochhammer terms in the two sums. The sum $\delta_3+\delta_5$ can be obtained from Equation (139) by exchanging $\nu_{i\bullet}$ and $\nu_{\bullet n}$ and accounting for the neglected common factor of $-\Delta\Phi^{(1)}\left(\overline{\nu}_{in}+1,\nu+1\right)$.

The sum of the terms for $\{\varepsilon,\eta_1,\eta_2\}$ is

$$\begin{aligned}S\left(\varepsilon+\eta_1+\eta_2\right)=&-\frac{\nu_{i\bullet}\nu_{\bullet n}\left(\nu+1\right)}{\nu}\Delta\Phi^{(1)}\left(\nu_{i\bullet}+1,\overline{\nu}_{in}+1\right)\Delta\Phi^{(1)}\left(\nu_{\bullet n}+1,\overline{\nu}_{in}+1\right)\\&+\left(\nu_{i\bullet}-\nu_{in}\right)\left(\nu_{\bullet n}-\nu_{in}\right)\left[\mathrm{F}_2\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)-\mathrm{F}_3\left(\nu_{i\bullet},\nu_{\bullet n},\nu_{in}\right)\right]\ ,\end{aligned} \tag{141}$$

and with the summations explicitly written out is

$$S(\varepsilon+\eta_1+\eta_2)=-\frac{\nu_{i\bullet}\nu_{\bullet n}(\nu+1)}{\nu}(\bar{\nu}_{in}-\nu_{i\bullet})(\bar{\nu}_{in}-\nu_{\bullet n})\sum_{k=1}^{\infty}\frac{(\bar{\nu}_{in}-\nu_{i\bullet}+1)_{k-1}}{k(\bar{\nu}_{in}+1)_k}\sum_{l=1}^{\infty}\frac{(\bar{\nu}_{in}-\nu_{\bullet n}+1)_{l-1}}{l(\bar{\nu}_{in}+1)_l}$$
$$+(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})\left\{\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\left[1-\frac{(\nu_{\bullet n}-\nu_{in}+1)_k}{(\nu_{i\bullet}+\nu_{\bullet n}-\nu_{in}+2)_k}\right]\right. \tag{142}$$
$$\left.-\sum_{k=1}^{\infty}\frac{(\nu_{i\bullet}-\nu_{in}+1)_k}{k(k+1)(\nu_{i\bullet}+\nu_{\bullet n}-\nu_{in}+2)_k}\sum_{l=1}^{\infty}\frac{1}{l(l+1)}\left[1-\frac{(\nu_{\bullet n}-\nu_{in}+1)_l}{(\nu_{i\bullet}+\nu_{\bullet n}-\nu_{in}+2+k)_l}\right]\right\}.$$

This can be rearranged into the combined double summation

$$S(\varepsilon+\eta_1+\eta_2)=(\nu_{i\bullet}-\nu_{in})(\nu_{\bullet n}-\nu_{in})\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\left(1-\frac{(\nu_{\bullet n}-\nu_{in}+1)_k}{(\bar{\nu}_{in}+2)_k}\right.$$
$$-\frac{(\nu_{i\bullet}-\nu_{in}+1)_k}{(\bar{\nu}_{in}+2)_k}\sum_{l=1}^{\infty}\frac{1}{l(l+1)}\left\{1-\frac{(\nu_{\bullet n}-\nu_{in}+1)_l}{(\bar{\nu}_{in}+2+k)_l}\right. \tag{143}$$
$$\left.\left.+\frac{\nu_{i\bullet}\nu_{\bullet n}(\nu+1)}{\nu(\bar{\nu}_{in}+1)}\frac{(k+1)(\bar{\nu}_{in}+k+1)}{(\nu_{i\bullet}-\nu_{in}+k)}\frac{(l+1)(\nu_{\bullet n}-\nu_{in}+1)_{l-1}}{(\bar{\nu}_{in}+1)_l}\right\}\right).$$

Equation (143) is symmetric under the exchange of $\nu_{i\bullet}$ and $\nu_{\bullet n}$, but numerical computation is faster and more accurate when the larger term is associated with $\nu_{i\bullet}$.

Implementation of the second functional revisions in MATLAB leads to the data plotted in Figure 11. For the code based upon this version of the functions, departure from the appropriate limit still begins to occur at a scale factor of 10^8, or a total count size around 10^{14}. This departure is significantly less dramatic than that seen in Figure 11. Through the examination of the scaling behavior of about 70 other combinations of counts, the numerical errors have been found to grow when $\nu_{i\bullet}>10^{11}$ or $\nu_{\bullet n}>10^{11}$. Because of this behavior, the MATLAB code that estimates the covariance values has an option to rescale the collection of counts so that $\nu_{i\bullet}\le 10^{11}$ and $\nu_{\bullet n}\le 10^{11}$ for specific values of i and n when calculating terms in $\sigma^2_{\overline{IN}}$. The four rescaled values are used to estimate the sub-terms of $\sigma^2_{\overline{IN}}$ in Equations (123) through (134), and the software accounts for the division by $(\nu+1)$ when estimating the sub-terms. A final division by the unmodified count ν is applied to the sum of sub-terms to get the contribution to $\sigma^2_{\overline{IN}}$ for the overall sum of i and n. This option is turned on by default in the software package and may be disabled by the user. The software issues a warning to the user if this scaling is invoked during execution. It is believed that the fractional error between the values calculated by the software with rescaling and the true values are small enough to be ignored in most applications. The final division of the scaled terms in $\sigma^2_{\overline{IN}}$ by the unmodified value of ν causes the overall value of $\sigma^2_{\overline{IN}}$ to continue to decrease in magnitude as the counts continue to increase.

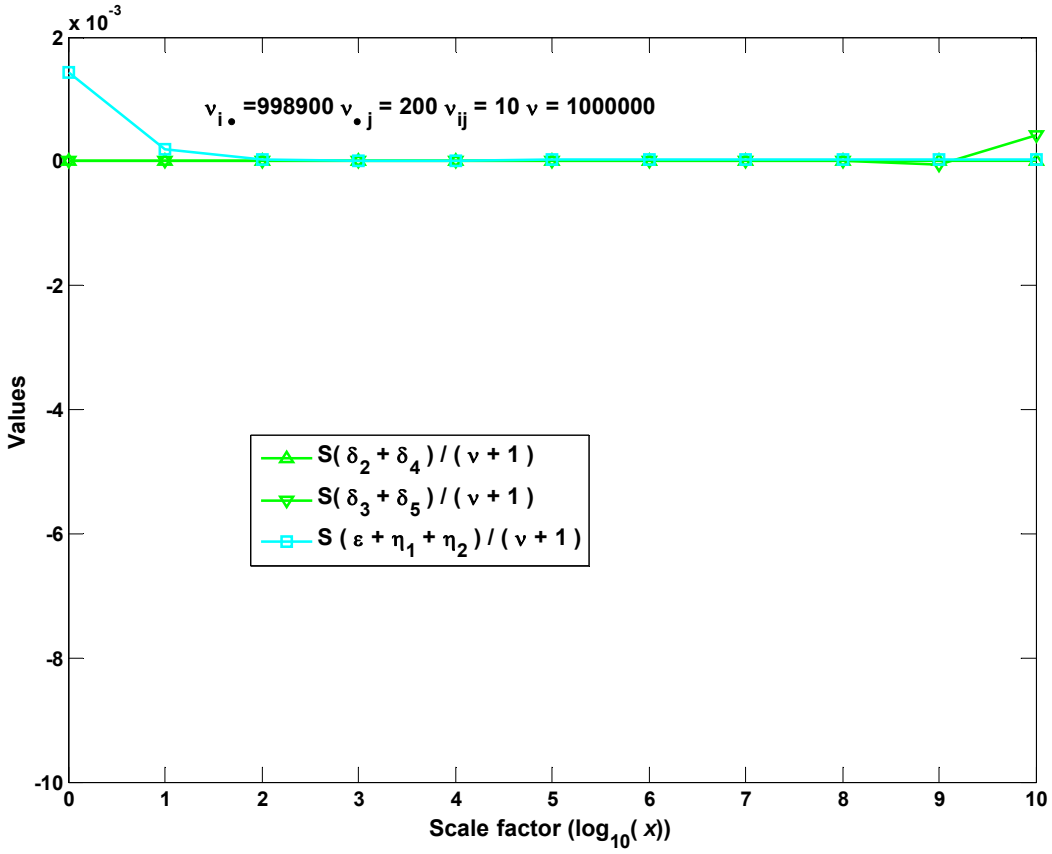


Figure 11. The combined and reformulated sub-terms of σ^2_{IN} *for different multiplier scale factors.*

Figure 12 shows the eigenvalues for the entropic covariance matrices produced by three different versions of the algorithm. The results were produced using the accumulation matrix in Equation (54). The eigenvalue calculations from the second revision of the algorithm are stable for large sample sizes that are scaled up from this test matrix. Part of this stability was achieved by scaling the counts in the estimation of σ^2_{IN} as the counts passed 10^{11}. This is not to say that other accumulation matrices will not suffer from numerical round-off problems, but that the estimates look reasonable for this example. It might be noted that at some point in increasing the sample size, the major inaccuracy in the estimated mean values of the information theoretic measures will be the round-off error, even with the use of double precision arithmetic with a precision of about 10^{-16}. The errors will be smaller than the double precision limits on the means, and researchers will have to account for the precision of the mean in addition to statistical errors.

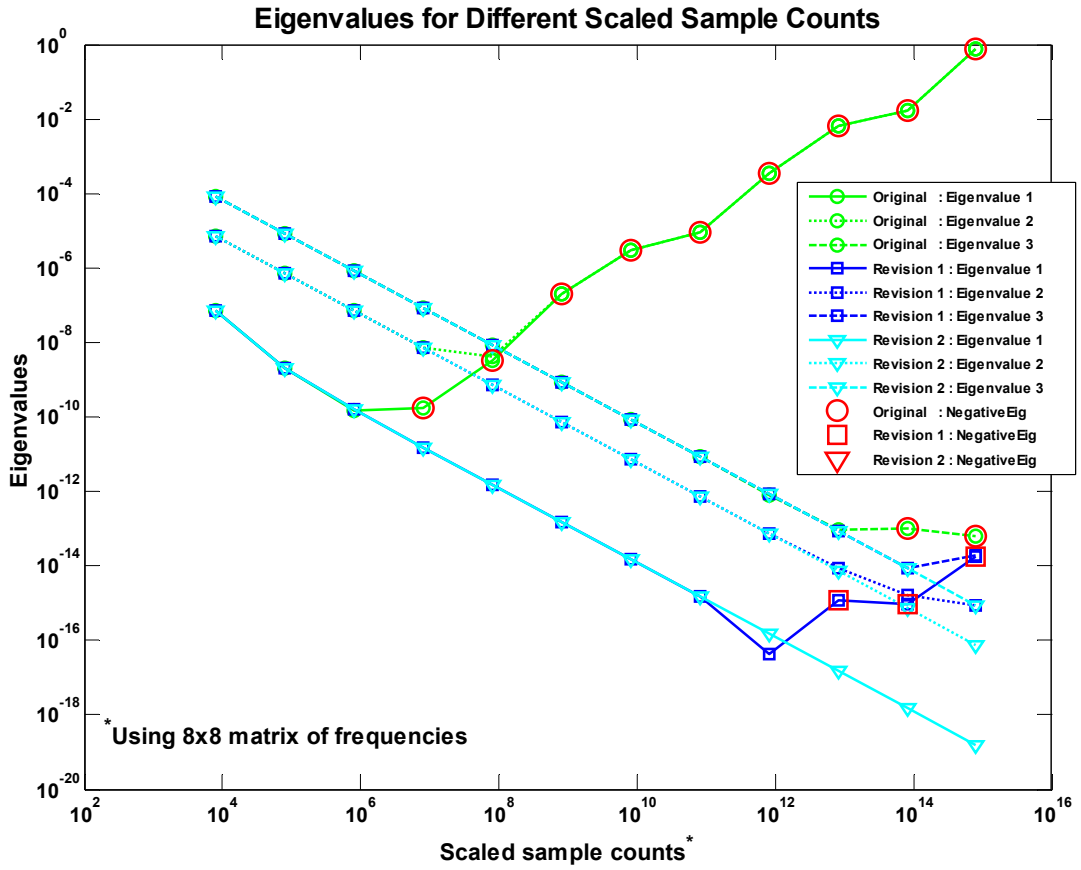


Figure 12. Eigenvalues for the original and two versions of the information theoretic algorithm. Revision 2 rescales the counts if the values are too large to avert numerical precision problems when estimating eigenvalues. Negative eigenvalues are shown with red circles, squares, and triangles to indicate that the covariance terms are incorrectly estimated.

The computation times for the three different algorithms are shown in Figure 13. The second revision runs with fairly consistent times across different scale factors for the counts in the 8×8 test matrix. The first revision is generally the fastest except for when the number of samples is around 10^9 to 10^{11}. Both revisions usually execute more quickly than the authors' original implementation of Wolpert and Wolf's equations. The second revision always executes faster than the original code for the example shown.

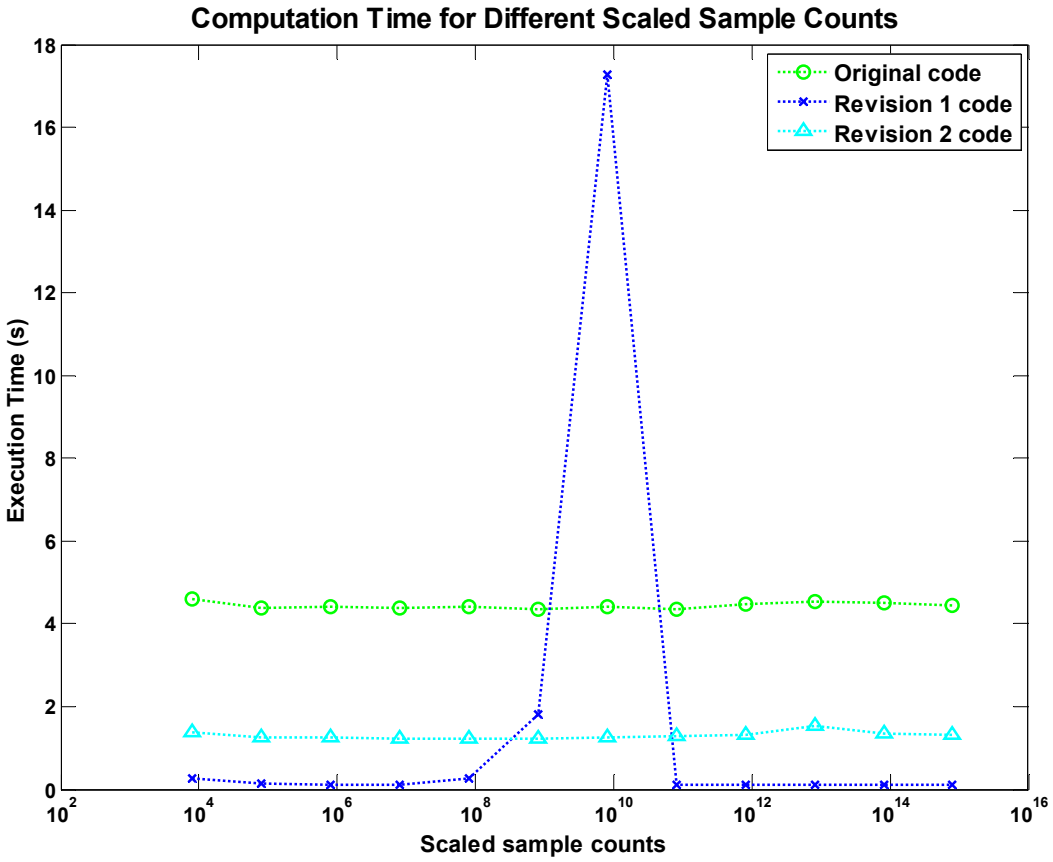


Figure 13. The computation times for different scale factors on a set of counts for three different algorithms for computing information theoretic terms.

The plot is for only one accumulation matrix. It is expected that even the second revision will show increases in estimation times like that of the first revision for other accumulation matrices. Users should not expect that the computer code associated with the second revision will execute with the consistent times shown in this plot.

8. RESULTS OF THE FINAL FORMULATION FOR TRACKING ALGORITHM EVALUATION

Once it was believed that reasonably stable estimates of covariance errors could be made for any sample size, an evaluation of the performance of two simulated tracks was performed. Figure 14 shows the results from the simulated data sets used to make Figure 1. The plotted error ellipses are 95% confidence regions.

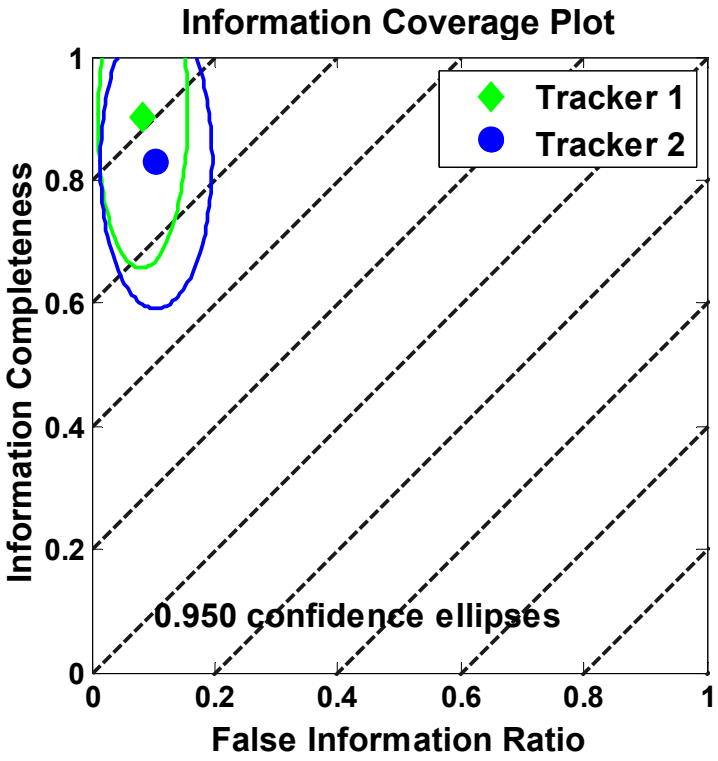


Figure 14. An information coverage plot for simulated track data generated with revision 2 of Wolpert and Wolf's equations.

An uninformed assessment of the plot would be that the data sample size is insufficient to confidently determine the best tracker. Part of the misleading nature of the plot is that the 95% confidence regions are in a two-dimensional space and therefore need to be quite large to enclose 95% of the probability densities. The ellipses do not provide an accurate visual indication of the confidence that one tracker is better than another when they overlap. Direct evaluation of total conditional entropy provides a

much better estimate of statistical significance than the two-dimensional data. This is partly because changes to the measurements that are parallel to the dotted lines in Figure 14 do not change the total conditional entropy and can be considered to be a nuisance dimension. Integration of the error ellipses to remove this nuisance dimension produces the data plotted in Figure 15 and leads to a more informative hypothesis test.

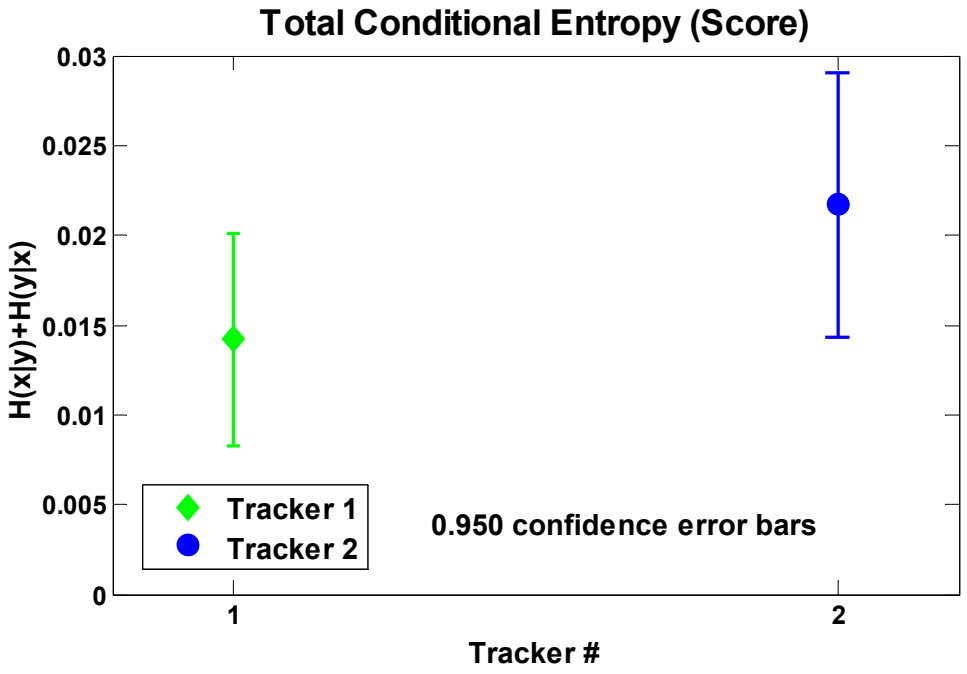


Figure 15. A plot of the total conditional entropy for two trackers, with error bars.

The proposed measure for the overall comparison of the relative performance between trackers is the total conditional entropy. Figure 15 shows the total conditional entropy for the two example trackers. Tracker 1 is considered to perform better than Tracker 2 because of the lower total conditional entropy. The total conditional entropy, with error estimates, for Tracker 1 is 0.0142 ± 0.0015 and for Tracker 2 is 0.0217 ± 0.0019. The probability that statistical effects may have led to the wrong conclusion can be calculated from the test statistic of Equation (11) and is 0.001. In other words, there is about 1 chance in 1000 that tracker two is really better than tracker one, given the sample statistics. This estimate should be considered in light of the caveats that the error estimates only incorporate contributions due to statistical effects and do not include any contributions from systematic effects. It also does not account for any biases due to the fact that the true distributions are non-Gaussian and the values are always non-negative. This is likely to be a second- or third-order effect in that both distributions will be skewed to the same

order. Even with the consideration of these caveats and possible biases, one might still be inclined to prefer Tracker 1 over Tracker 2. Analysis presented later in the report on the sensitivity of the information theoretic measures to parameters in the analysis software may incline one to make a different choice.

In the event that the power of the test (as evidenced by the statistics) is insufficient to draw a strong enough conclusion as to which is the best tracker, a larger data set may be acquired for testing or the results of multiple independent data sets may be combined. The information theoretic measures, $H(x)$, $H(y), H(x,y), H(x \mid y)$, and $H(y \mid x)$, are additive for independent systems and provide a convenient method for increasing statistical support. The information theoretic measurements and errors from multiple independent tests can be combined and the resultant sums used to assess the statistical significance of the differences in total conditional entropy. With simulations of tracker performance, it may be possible to estimate the expected statistical power that a data set will provide, given the number of tracks and their lengths.

8.1 SENSITIVITY TO ASSOCIATION THRESHOLDS

The technique described in this report and contained in the software package has two parameters that can affect the estimated information theoretic measures. The first parameter is the total number of counts that the accumulation matrix contains. The second applies to the algorithm that associates estimated tracks to truth tracks over a number of time frames. There is an association threshold parameter used to determine when an estimated track should be associated with a truth track or considered to be unassociated with any truth track. These parameters are not required for the construction of association matrices for classifiers.

The software package contains a function that performs the track-to-track association using a probabilistic cost function and a function that performs the linear assignment of estimated tracks to truth tracks. Other researchers may wish to use their own association cost function and assignment algorithms to construct their accumulation matrices. The sensitivity of the information theoretic results to changes in the association threshold and total count size are examined next. Results should be qualitatively similar to other cost functions and association algorithms.

Figure 16 shows the information coverage plot, parameterized for two different simulated trackers as a function of varying association threshold. As the association threshold is increased, more sensor tracks associate with the truth tracks causing measured performance to generally improve for both trackers. Tracker 1 is apparently better than Tracker 2 in this plot for all values of the association thresholds that were tested. The figure shows that the error covariances tend to increase slightly as the thresholds are increased. The error ellipses do not provide strong visual evidence for Tracker 1 being declared significantly better than Tracker 2.

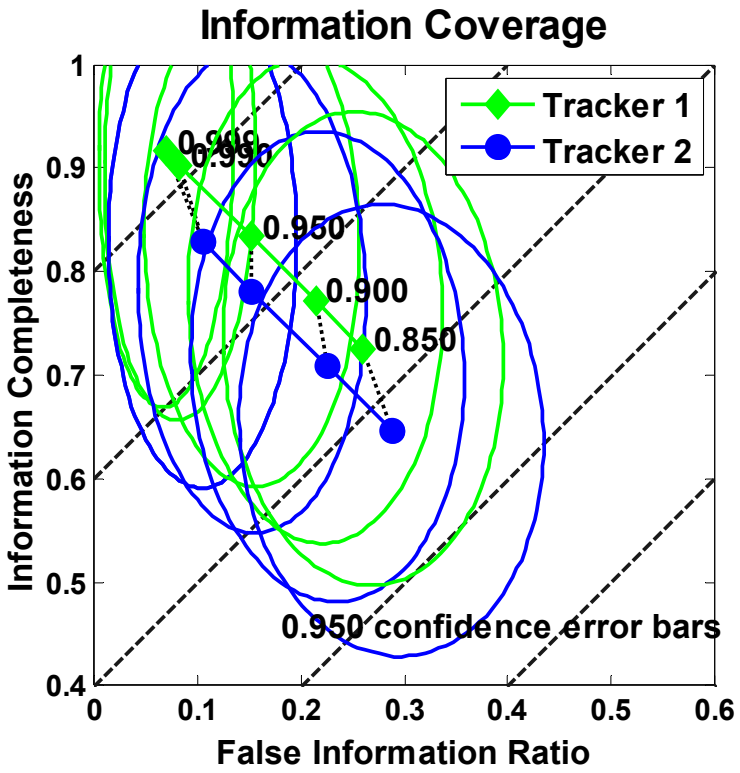


Figure 16. The information coverage plot for the two simulated trackers for different values for the association threshold that is used to construct the accumulation matrix used to calculate the information theoretic measures. 95% confidence error ellipses are shown.

Figure 17 shows the total conditional entropy for different values of the association threshold. Total conditional entropy decreases as the association threshold is increased, as would be expected from the data plotted in Figure 16. The error bars that are plotted are the 95% confidence intervals and assume that the distributions are Gaussian. Again, Tracker 1 is shown to be performing better than Tracker 2 for all tested thresholds. Again, the naïve interpretation of the error ellipses would imply that there is weak statistical evidence that Tracker 1 is better the Tracker 2. Evaluation of errors associated with total conditional entropy provides a better assessment of which tracker is better.

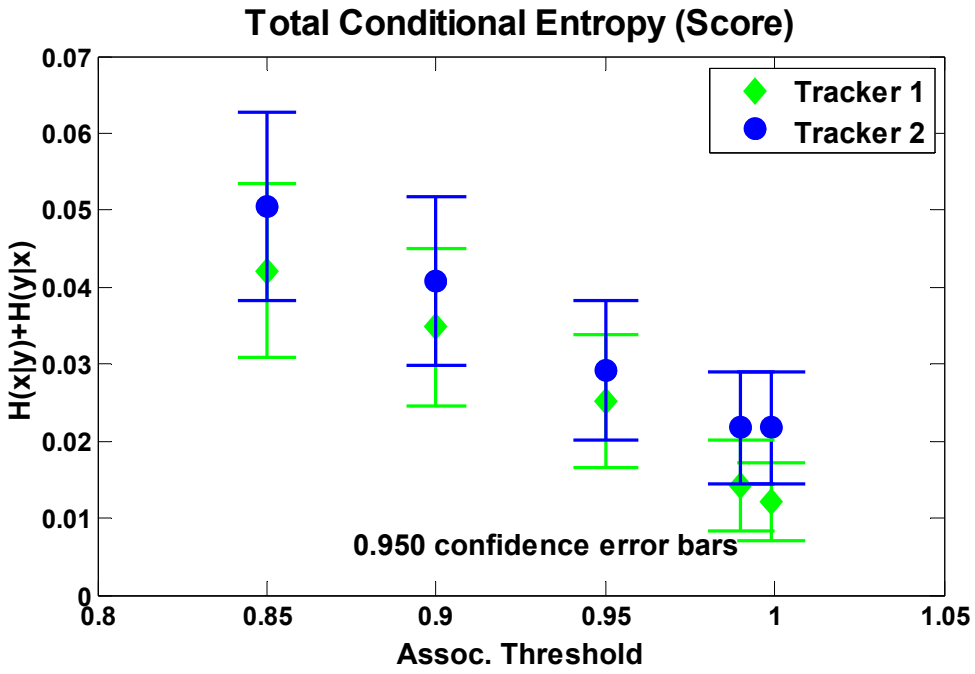


Figure 17. The total conditional entropy for the two simulated trackers for different values for the association threshold that is used to construct the accumulation matrix used to calculate the information theoretic measures. 95% confidence error bars are shown.

Table 2 shows the values plotted in Figure 17, as well as the probability that an incorrect decision may have been made as to which is the better tracker. The error values are the one-standard deviation errors. For the data points that are evaluated, the decision error peaks at a value of 0.103, or about 10%, for an association threshold value of 0.95. The decision error decreases significantly as the association threshold is increased. This suggests that some care should be taken when selecting the association threshold for testing trackers so that the thresholds accurately reflect the true nature of the data sets and tracking systems, especially with regard to the probabilities for missed detections and false tracks. It may be worth analyzing the sensitivity to the association threshold to support the decision to select one tracker over another. It is highly unlikely that the decision for the better tracker will switch as the threshold is changed. An accurate assessment of the probability of a wrong decision may be important information in the selection of a tracker or in the decision to evaluate the trackers with larger data sets.

TABLE 2

Total Conditional Entropy and Probability Estimates of Decision Error for Different Association Thresholds

Association Threshold	Tracker 1 Total Conditional Entropy	Tracker 2 Total Conditional Entropy	Probability of Decision Error
0.850	0.0421 ± 0.0029	0.0505 ± 0.0032	0.026016
0.900	0.0348 ± 0.0027	0.0408 ± 0.0029	0.064873
0.950	0.0251 ± 0.0023	0.0293 ± 0.0024	0.102866
0.990	0.0142 ±0.0015	0.0217 ± 0.0019	0.001059
0.999	0.0121 ± 0.0013	0.0217 ± 0.0019	0.000020

8.2 SENSITIVITY TO THE TOTAL NUMBER OF COUNTS

A similar sensitivity analysis can be performed for the total number of counts in the accumulation matrix. For trackers, this is usually estimated from information about the state space of the data sets being used in the evaluation. Each dimension in the state space of the test data set contributes a multiplicative factor to the overall count. The contribution of quantized dimensions in the state space is the product of the number of elements for those dimensions. For real-valued dimensions, the contribution is determined from the ratio of the ranges of the dimensions versus the desired resolutions in those dimensions. The product of the quantized numbers for all the dimensions estimates the total number of counts that the accumulation matrix should contain. In practice, the accumulation matrices are filled by comparing truth and sensor measurements to populate the bins. The (1, 1) bin is filled with the difference between the total number of counts estimated for the state space and the total number of counts entered in all the other bins in the array. This causes the total number of counts in the entire matrix to equal the total number of possible distinguishable states in the state space of the test data set. For classifiers, the total number of counts is entirely set by the size of the data sample. Larger data samples may be required to obtain enough statistical significance for a test.

The sensitivity analysis presented here varies the total number of counts for the simulated data set which hypothetically generated the two track data sets used as examples in this report. The association threshold for this sensitivity analysis was selected to be a constant value of 0.95 because, in the previous sensitivity analysis, this produced the highest probability that Tracker 2 could be better than Tracker 1 (for the values evaluated). The information theoretic measurements and errors were estimated for different values for the total number of counts. The total number of counts were scaled by factors of 1/10, 1, and 10, and the accumulation matrices for Trackers 1 and 2 adjusted to produce the desired counts. In all cases, the (1, 1) bin in the accumulation matrix was modified to get the desired total number of counts.

Given an original total number of counts of 40,000, the three analyses were conducted for accumulations matrices with 4,000, 40,000, and 400,000 counts. Figure 18 shows the information coverage plot for the three different scales of the total number of counts. As the total number of counts increases, the information completeness increases and the false information ratio decreases. The error covariances appear to increase slightly as the total number of counts increases.

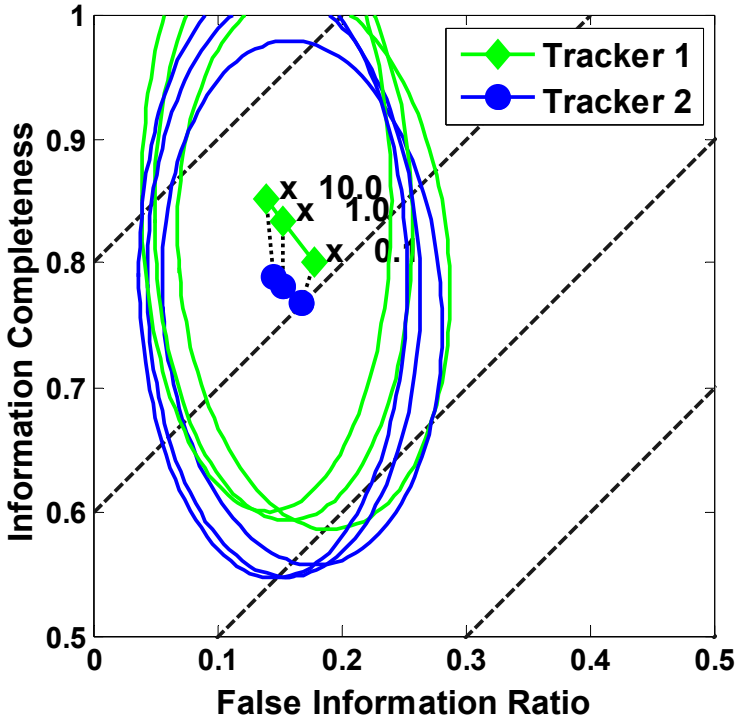


Figure 18. The information coverage plot for two simulated trackers with different multipliers on the total number of counts in the accumulation matrix.

Figure 19 shows the total conditional entropy for the three different scale factors. A log-log plot is used because of the range of scale factors and because the total conditional entropy approximately scales as the inverse of the total number of counts. The difference in performance between the two trackers widens as the scale factor increases.

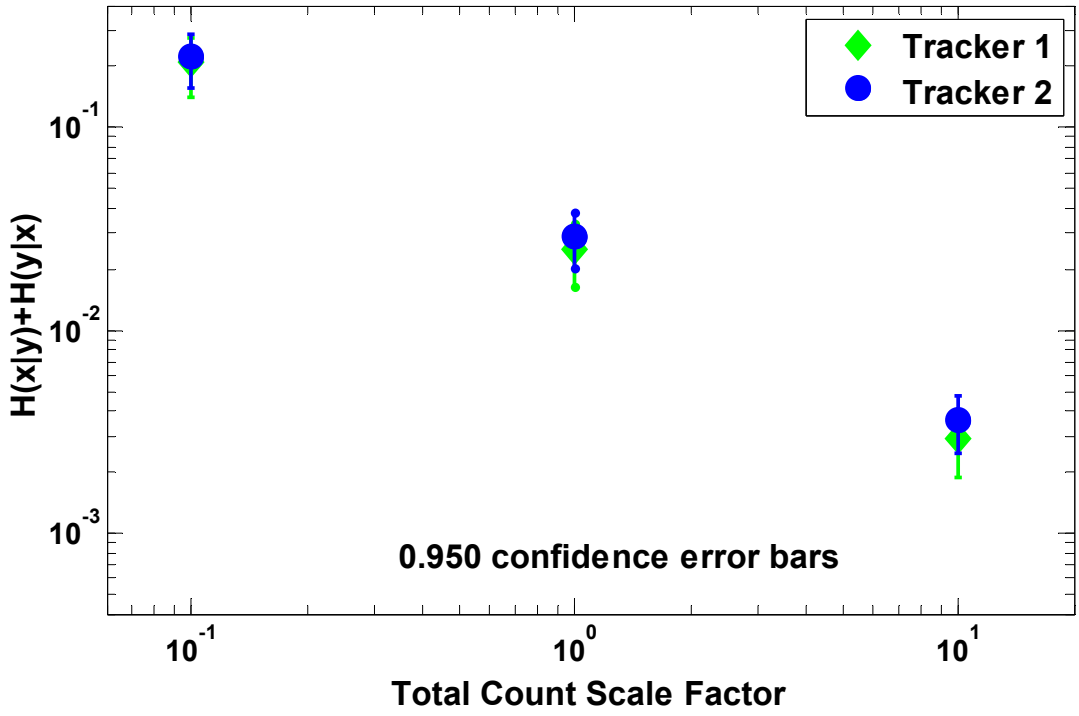


Figure 19. The total conditional entropy for the two simulated trackers for different scales on the total counts in the accumulation matrix. The original total number of counts was 40,000.

Table 3 shows the total conditional entropy for the two trackers, with one-standard deviation errors for three different scale factors on the total number of counts. The decision error probability for selecting the poorer performing tracker as the better tracker is also shown. The probability decreases as the total number of counts increases by approximately the same order of magnitude. The results imply that accurate estimates of the total number of counts for the data sets are required to generate relatively accurate estimates of the decision error probabilities. A sensitivity analysis can be performed if there are doubts on the appropriate choice for total number of counts. The data set can also be expanded to increase the statistical weight of the evaluation. For this example, if a decrease in the association threshold to 0.95 produces a decision error probability of about 10% and with a further decrease in the expected total number of counts by a factor of 1/10 produces a decision error probability of 30%, a cautious researcher who cannot defend more stringent limits for the association threshold and total number of counts might elect to process additional data sets to drive down the errors so that the better tracker can be confidently identified.

TABLE 3

Total Conditional Entropy and Probability Estimates of Decision Error for Different Scales of Total Number of Counts

Total Count Scale Factor	Tracker 1 Total Conditional Entropy	Tracker 2 Total Conditional Entropy	Probability of Decision Error
0.1	0.20926 ± 0.01756	0.22247 ± 0.01715	0.295211
1.0	0.02514 ± 0.00226	0.02928 ± 0.00237	0.102866
10.0	0.00292 ± 0.00027	0.00361 ± 0.00030	0.044895

With the availability of errors on the information theoretic measures, it is now possible to determine the significance of the results of tracker and classifier performance analyses. It is also possible to determine the sensitivity of the analyses to parameters and algorithms in the assessment code itself. The provided examples of sensitivity analyses show that these influences can have a significant influence on the significance of the results. The selection of the association threshold and total number of counts parameters should be motivated by sound reasoning so that analysis results are reliable. If the estimated accuracies of these parameters are poor, a sensitivity analysis should be performed to ensure that the better tracker is identified with an acceptable level of confidence. If confidence is low, additional data sets can be analyzed to drive down the statistical errors.

This page intentionally left blank.

9. SOFTWARE FOR THE EVALUATION OF TRACKING AND CLASSIFICATION ALGORITHMS

A software package has been developed in MATLAB for other researchers to use in their evaluation of tracking and classification algorithms. The package contains a directory tree: the top folder, which contains a pdf file of a paper from the International Conference on Computer Vision 2009 [1], a pdf file of this technical report, and two folders: 'code' and 'truthData.'

The folder 'truthData' contains a MATLAB file, testdata.mat, which contains test data that are used by test functions in the 'code' folder to check that the algorithms are running correctly. It also contains the file dataGenerator.m, which was used to generate testdata.mat. This file generates a simple, contrived set of simulated truth and tracker generated tracks similar to what might be generated by a real tracking algorithm. This data set was used to produce the information coverage plots, significance tests, and sensitivity analyses in this document. Various deficiencies are intentionally placed into the simulated tracker tracks. The file can also be used as a template to construct data structures from other truth data sets and trackers so that the function calls that associate the sensor and truth tracks and generate information theoretic algorithms can be more easily called.

The folder 'code' contains the file readme.txt and eight MATLAB .m files:

associateSystemAndTruthTracks.m
assocLapmod.m
computeMetricsFromAccumMat.m
Ellipse_plot.m
lapmodAssocSparse.m
plotTrackerEvalResults.m
s10Wolpert_Wolf.m
trackerEvalTestScript.m

The file readme.txt contains information to help users to begin using the software package.

The key MATLAB function for the user is computeMetricsFromAccumMat.m. This function estimates the information theoretic measures from accumulation matrices. The function associateSystemAndTruthTracks.m may be useful for those users intending to evaluate trackers. The function plotTrackerEvalResults.m may be useful for plotting tracker metrics. The function trackerEvalTestScript.m is provided to test the software. This function should be run from the 'code' directory.

The other MATLAB files are support functions, and most users will not need to concern themselves with the details in the function. The function s10Wolpert_Wolf.m contains the code that generates information theoretic parameters from accumulation matrices. The code in this function is derived from

the equations contained in this document. The MATLAB functions assocLapmod.m and lapmodAssocSparse.m contain code to execute a linear assignment algorithm based upon an algorithm developed by Volgenant [14].

9.1 USE OF COMPUTEMETRICSFROMACCUMMAT.M

The function, computeMetricsFromAccumMat.m, accepts up to three input variables:

accumMat – A matrix containing the counts of the number of times that the classes of two sets of objects were associated. The matrix must contain non-negative real values.

prior – [Optional] The value of the prior distribution factor to apply to accumMat. This may be entered as a string: 'Haldane,' 'Perks,' 'Jeffrey,' 'Bayes,' 'uniform' (default), a non-negative real value, or a non-negative real matrix that is the same size as accumMat.

doCovs – [Optional] A flag to cause the function to calculate and return a covariance matrix, if true. The default is false.

For the evaluation of multi-target trackers, the accumulation matrix, *accumMat*, is constructed so that the first row and column store the counts of the number of times that truth tracks or system tracks did not associate with a corresponding track in the other set. The (1, 1) element is filled with a count of the number of true negatives. This value is estimated by quantizing the state space used to associate the two sets of tracks. The total number of counts in the accumulation matrix is estimated from the total practical volume of the state space divided by the typical resolution of the states for the two sets of tracks. The sum of all cells in *accumMat* is expected to equal this number, and cell (1, 1) is set to satisfy this condition.

For the evaluation of classifiers, the accumulation matrix, *accumMat*, is constructed so that each row and column corresponds to the number of counts where the data for truth class i was associated with decision class j.

The input variable, *prior*, defaults to uniform or Bayes' prior. Strings can be entered to select the kind of prior that a user wishes to adopt for their calculations. The string 'Haldane' will add zero (0) to all the entries in the accumulation matrix. The string 'Bayes' or 'uniform' will add one (1) to all entries in the matrix. The string 'Jeffrey' will add 1/2 to all the entries in the matrix. The string 'Perks' will add 1/(N×M) to all the entries in the matrix, where N and M are the sizes of the matrix dimensions. Users can provide a non-negative real value that will be added to all entries in the matrix accumMat or a non-negative real matrix of priors that will be added to the matrix accumMat. The matrix of priors must be the same size as the accumulation matrix.

The Perks' and Haldane's priors have special importance in information theory because only the Perks' prior and the Haldane's prior will result in the same calculations of the entropies, H(x) and H(y), for different numbers of dimensions. For example, if one experimenter has access to the two-dimensional accumulation matrix and another experimenter only has access to the counts associated with one of the

dimensions, they can only be guaranteed to calculate the same entropy values for the common dimension if they both use either Haldane's prior or Perk's prior. All other priors will in general cause the two experimenters to calculate different values. Haldane's prior leaves the prior adjusted counts unchanged, so it is obvious why the entropy calculations are consistent. Perk's prior is different and a short explanation can clarify why this is the case. For the Perks' prior, the first experimenter will add 1/(N × M) to each two-dimensional cell, while the second experimenter will add 1/N to each one-dimensional cell. When the first experimenter calculates $\nu_{i\bullet}$ for use in Equation (16) , his summation over M will produce values equal to the second experimenter's values for ν_i, thus producing the same estimates for the entropy. This is especially useful for the evaluation of trackers, because different trackers will usually generate different numbers of tracks and accumulation matrices of different sizes. In general, the selection of other priors will cause the estimates of the truth entropy to differ between the two experimenters [15].

It is recommended that the input variable *dCovs* be allowed to default to 'false' when preliminary calculations are being generated. Set the flag to 'true' when covariance matrices are actually needed. The calculation of the covariance matrices can take a significant amount of time to complete, depending upon the number of counts and the size of the matrix, *accumMat*.

The output variable is a structure:

result – result structure with the following fields:

.infoCompleteness	– Information completeness
.falseInfoRatio	– False information ratio
.truthPurity	– Truth purity (also known as one-to-one completeness)
.trackPurity	– Track purity
.truthCompleteness	– Truth completeness (also known as many-to-many completeness)
.trackCompleteness	– Track completeness
.trackLenHist	– The associated track length histogram (in frames)
.means	– The means of the information theoretic measures
.sList	– Text describing the contents of the variable 'means'
.covs	– The covariance matrix for the means, it is returned if doCovs flag is 'true'

The field *infoCompleteness* is the ratio of the expectation of mutual information versus the expectation for the truth entropy:

$$I_C = E\left(I\left(x;y\right)\right)/\,E\left(H\left(x\right)\right)\,, \tag{144}$$

where

$$I(x;y) = H(x) + H(y) - H(x,y)$$

so

$$E\left(I\left(x;y\right)\right) = E\left(H\left(x\right) + H\left(y\right) - H(x,y)\right). \tag{145}$$

The field, *falseInfoRatio*, is the ratio of the expectation of the conditional entropy of the system data given the truth data versus the expectation of the truth entropy:

$$R_{FI} = E\left(H\left(y \mid x\right)\right) / E\left(H\left(x\right)\right). \tag{146}$$

The form of the equation for the purity measure for individual system track associated with column j is

$$P_j = \frac{\max\left(\nu_{2j}, \nu_{3j}, \ldots, \nu_{Mj}\right)}{\sum_{i=1}^{M} \nu_{ij}}, \tag{147}$$

where ν_{ij} is the accumulation matrix. The maximization over the indices can be swapped between i and j to obtain individual truth track purity measures for truth tracks associated with the corresponding rows in the accumulation matrix.

The overall purity for the field *trackPurity* is calculated with

$$P = \frac{1}{N} \sum_{j=2}^{N} \frac{\max\left(\nu_{2j}, \nu_{3j}, \ldots, \nu_{Mj}\right)}{\sum_{i=1}^{M} \nu_{ij}}, \tag{148}$$

where the indices for the sum and maximization function are swapped to obtain the field *truthPurity*.

The form of the equation for the completeness measure for the individual system track associated with column j is

$$C_j = \frac{\sum_{i=2}^{M} \nu_{ij}}{\sum_{i=1}^{M} \nu_{ij}}. \tag{149}$$

The indices for the sums are swapped to produce the corresponding measure for individual truth tracks. The equation for the overall completeness field, *trackCompleteness*, is

$$C = \frac{1}{N} \sum_{j=2}^{N} \frac{\sum_{i=2}^{M} \nu_{ij}}{\sum_{i=1}^{M} \nu_{ij}}. \tag{150}$$

The indices over the sums are again swapped to produce the statistics for the field *truthCompleteness*.

The field *trackLenHist* is a histogram of the lengths of the tracks provided by the tracker. This is effectively the number of counts in each column of the accumulation matrix. Some researchers consider track length to be a useful indicator of tracker performance.

The field *means* is a 1 × 7 vector of all the information theoretic quantities that are calculated by the function, computeMetricsFromAccumMat.m. The values returned in *means* are the total entropy, $H(x,y)$, the truth entropy, $H(x)$, the tracker entropy, $H(y)$, the mutual information, $I(x;y)$, the truth conditional entropy, $H(x \mid y)$, the sensor conditional entropy, $H(y \mid x)$, and the total conditional entropy, $H(x \mid y)+H(y \mid x)$.

The field *sList* is a cell array of strings defining the variables in *means*. It is intended to be used to define axis labels when plotting data in MATLAB.

The field *covs* is a 7 × 7 covariance matrix for the vector *means*. It should be remembered that there are only three independent variables in the 7 × 7 matrix. The estimation of eigenvalues and eigenvectors for covariance matrices will only return meaningful results if three or fewer rows and columns are selected for the operation.

9.2 USE OF ASSOCIATESYSTEMANDTRUTHTRACKS.M

The function associateSystemAndTruthTracks.m is used to construct the accumulation matrix that is used as input into computeMetricsFromAccumMat.m. It is used to evaluate the performance of multi-target trackers and is not used with classifiers. It establishes a frame-by-frame association between system tracks and truth tracks and populates the accumulation matrix by counting the number of times that truth and system tracks are, or are not, assigned to each other.

function accumMat = associateSystemAndTruthTracks(systemTracks, truthTracks, stateSpaceSize,
removeUnassociatedSystemTracks, confidence)

Input:

systemTracks – An array of tracks from the tracker indexed by sample frames.

Required fields: (m tracks in this frame)

.id (m)	– track ids
.mean (m × d)	– track states (d states)
.cov (m × d × d)	– track covariance

truthTracks – An array of truth tracks indexed by sample frames.

Required fields: (n tracks in this frame)

.id (n)	– track ids
.mean (n × d)	– track states (d states)
.cov (n × d × d)	– track covariance

stateSpaceSize – The size of the state space (the number of distinguishable objects that fit in the state space). This is typically a very large number.

removeUnassociatedSystemTracks – A flag to indicate whether to remove unassociated system tracks. This is useful when the evaluator has incomplete truth and wants to avoid penalizing the observation of untruthed tracks (optional: default = false).

confidence – The association gating parameter (optional: default = 0.99). Valid values range from 0 to 1. The inverse chi-squared function is used to set the internal threshold and accounts for the dimensionality of the state space (increasing confidence leads to a larger gate and looser associations).

Output:

accumMat – A (w+1 × z+1) dimensional matrix, where w is the total number of truth tracks and z is the total number of system tracks. The first row contains the counts of the number of times each sensor track failed to associate with any truth tracks (false system tracks). The first column contains the counts of the number of times each truth track failed to associate with any system tracks (missed truth tracks). Cell (1, 1) contains the total number of true negative counts and the function uses the input variable, stateSpaceSize, to estimate this number.

The function assumes that the *systemTracks* and *truthTracks* are sampled at the same times and have the same array size (i.e., number of sample frames). It is assumed that the state spaces of the system and truth tracks are consistent.

This function calls assocLapmod.m, which in turn calls lapmodAssocSparse.m to perform the truth-to-tracker association with a linear assignment algorithm that associates the tracks based upon their covariance weighted Mahalanobis distances and the association confidence value.

9.3 CODING TECHNIQUES IN S10WOLPERT_WOLF.M

Although the casual user may not be interested in the details of s10Wolpert_Wolf.m, some details are provided for more serious users who may wish to validate or improve on the accuracy of the covariance estimates that are returned by computeMetricsFromAccumMat.m.

The functions for $\Delta\Phi^{(1)}(x,z)$ and $\Delta\Phi^{(2)}(x,z)$ have been coded as special algorithms to reduce numerical precision errors. The MATLAB psi function is somewhat inaccurate for estimating differences when both *x* and *z* are large and close to each other. When *x* and *z* are within 90% of each other and less than 10^8, the iteration rule of Equation (61) is used to sum terms until the difference between *x* and the modified *z* are less than one (1). The MATLAB psi function is then used to account for the fractional differences between the two modified values. Otherwise, the MATLAB psi function is used to directly estimate $\Delta\Phi^{(n)}(x,z)$. Equivalent summations are performed for $\Delta\Phi^{(2)}(x,z)$ under the same conditions. The code can be found in s10Wolpert_Wolf.m and the recursion relationships in Abramovitz and Stegun [12].

The terms that involve summations that contribute to the covariance term, $\sigma^2_{\overline{IN}}$, are implemented as recursive functions. The terms are grouped into pairs of sub-terms with a common factor of 1/(k*(k + 1)) to maximize the cancellation between sub-terms. Common constant terms in the two sub-terms are moved outside of the summation, causing the first sub-term within the summation to have a value of one (1) and the second sub-term to be a function of gamma function ratios and other multiplicative terms. The functions are written to recursively call themselves when the value for the next term in the summation drops below a defined percentage of the accumulated sum of terms at the current level in the recursion. The called function begins to sum another block of terms. When the second sub-term drops below an internally defined value (currently the double precision epsilon value in MATLAB, ~2.2e-16), the recursive call terminates, and the collected partial sum at each level in the recursion is combined with the returned sum from the lower level to compute the overall sum. The recursive summation significantly improves the accuracy of the overall result. The recursive calls also return the number of summations that were performed. When the summation terminates, the first term (with a value of one) totally dominates the uncompleted summation to infinity. Its contribution can be accounted for by an analytical solution to the sum of 1/(k*(k + 1)) from q to infinity using

$$\sum_{k=q}^{\infty} \frac{1}{k(k+1)} = \frac{1}{q}. \tag{151}$$

The recursive summation reduces the accumulated round-off to some degree while still allowing the gamma ratio terms to be calculated as simple multiplications with previously accumulated gamma ratio terms.

If the recursion level reaches an internally defined depth without the second term dropping below the termination value, the recursive function calls an approximation function that interpolates terms across multiple step sizes. The step size is continually increased such that the function call will terminate after a reasonable number of steps. The function also terminates if the second term drops below the internally defined value (currently the double precision epsilon value in MATLAB, ~2.2e-16).

To interpolate the second term in the approximation function if the recursive call goes too deep, internal functions are used to calculate the double gamma ratio terms instead of using recursive multiplication. The approximation for gamma ratio terms is taken from Abramovitz and Stegun [12],

although other approximations are now available that could be more advantageous if more speed or accuracy is required [16–18].

The portion of the code in s10Wolpert_Wolf.m that calculates $\sigma^2_{\overline{IN}}$ constructs an array of values for the unique sets of $\nu_{i\bullet}$, $\nu_{\bullet n}$, and ν_{in} for all the terms in the accumulation matrix. The function swaps $\nu_{i\bullet}$ and $\nu_{\bullet n}$ to have the largest term first because the equation is symmetric in the exchange of these terms. Numerical computation is more accurate with the largest term first in the list. The unique list, with potentially swapped $\nu_{i\bullet}$ and $\nu_{\bullet n}$ terms, is then used to estimate the contribution of each set of elements to avoid identical computations. This dramatically speeds up execution of the code for highly symmetric accumulation matrices.

10. CONCLUSION

Error estimates on information theoretic measures are required to assess the statistical significance of tests on the relative performance of multi-target trackers and classifiers. The papers of Wolpert and Wolf provide exact formulae to estimate information theoretic measures and errors from data samples. Direct conversion of the equations into computer code provided good estimates of the information theoretic means and errors for sample sizes on the order of thousands, but became inaccurate for larger sample sizes. Reformulation of the equations to take advantage of cancellation between sub-terms in the equations has significantly improved the numerical precision of the computer code. The revised code provides a reasonable level of accuracy for all data sample sizes and should meet the accuracy needs of most researchers who evaluate multi-target trackers and classifiers. The computer code makes it possible to obtain a first-order estimate of the significance of analysis results and to determine if more data may be needed to support the conclusions of an analysis. The computer code may be useful to researchers who calculate information theoretic measures from data samples for other purposes besides decision system assessments.

This report has focused on methods to estimate reasonably accurate numbers for the information theoretic mean and covariance error terms. The study has proceeded with the assumption that the samples used to construct the accumulation matrices are statistically independent. Statistical independence of data samples for the analysis of classification algorithms can be achieved by taking care to select independent test samples. Statistical correlation in the analysis of tracking algorithms is much more difficult to avoid. The analysis method relies on the fact that tracker pathologies like track swaps, missed tracks, and extra tracks will spread counts across the accumulation matrix and increase the total conditional entropy. When a tracker is performing well, the estimated states for one time frame will be highly correlated with the next frame, and lead to some degree of coupling in the counts in the accumulation matrix. Additional research is required to identify how to quantitatively account for the influence of this coupling so that the significance of the test results can be accurately determined. While appropriate procedures are identified, one technique to reduce the impact of this systematic coupling is to use multiple, independent regions to assess tracker performance. The information theoretic measures from each region will be statistically independent and can be added together to get an overall performance assessment with greater accuracy and significance.

An additional area where effort could be devoted is in modifications to the software package to improve the execution speed. Conversion of the computer code from MATLAB to C or Java should significantly improve execution time. If required by some applications, additional speed could be achieved by conversion of the algorithm to execute on parallel processors. This might be useful for estimating information theoretic measures for very large dimension accumulation matrices. Other tweaks in performance can be made by modifying the constants used to terminate loops and initiate recursive calls, thereby trading numerical precision for speed. These trades are likely to be application dependent.

As a final note, an interesting discovery made during the course of improving the numerical accuracy of Wolpert and Wolf's equations was that the use of Perks' prior or Haldane's prior causes the algorithm to produce consistent estimates of information theoretic measures for different dimensions [15]. The entropy values calculated with Wolpert and Wolf's formulae for two dimensions produce the same values as those produced if only one dimension was used for the calculation of the entropies. This means that consistent information theoretic measures will be calculated with Perks' and Haldane's priors regardless of the number of nuisance dimensions that might be included in the calculations.

REFERENCES

1. E.K. Kao, M.P. Daggett, and M.B. Hurley, "An information theoretic approach for tracker performance evaluation," *IEEE 12th International Conference on Computer Vision*, 1523–1529 (2009).

2. R.S. Holt, P.A. Mastromarino, E.K. Kao, and M.B. Hurley, "Information theoretic approach for performance evaluation of multi-class assignment systems," *Proc. SPIE* 7697: 76970R/1-12 (2010).

3. D.H. Wolpert and D.R. Wolf, "Estimating functions of probability distributions from a finite set of samples," *Physical Review E* 52, 6841–6854 (1995).

4. D.H. Wolpert and D.R. Wolf, Erratum: "Estimating functions of probability distributions from a finite set of samples," *Physical Review E* 54, 6973 (1996).

5. D.R. Wolf and D.H. Wolpert, Los Alamos National Laboratory Report No. LA-UR-93-833 (1993) (unpublished) ~ Send email to comp-gas@xyz.lani.gov with subject "get 9403002" to get an encoded postscript version. "9403001" might also be helpful to the reader.

6. W. Perks, "Some observations on inverse probability including a new indifference rule," *J. Inst. of Actuaries Students' Soc.* 73, 285–334 (1947).

7. M. Hutter and M. Zaffalon, "Distribution of mutual information from complete and incomplete data," *Comp. Stat. & Data Anal.* 48, 633–657 (2005).

8. T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley-interscience (2006).

9. D.B. Owen, "A table of normal integrals," *Communications in Statistics – Simulation and Computation* 9:4, 389–419 (1980).

10. http://en.wikipedia.org/wiki/Pochhammer_symbol (2012).

11. A. Farhadi, M. Tabrizi, I. Endres, and D. Forsyth, "A latent model of discriminative aspect," *IEEE 12th International Conference on Computer Vision*, 948–955 (2009).

12. M. Abramovitz and I.A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York (1972).

13. K.B. Oldham et al., *An Atlas of Functions, Second Edition*, DOI 10.1007/978-0-387-48807-3_45, Springer Science+Business Media, LLC (2009).

14. A. Volgenant, “Linear and semi-assignment problems: A core oriented approach,” *Computers & Operations Research*, Vol. 23, No. 10, pp. 917–932 (1996).

15. Simon DeDeo, private communications (2011).

16. F. Qi, “Bounds for the ratio of two gamma functions – From Wendel’s and related inequalities to logarithmically completely monotonic functions,” arXiv:0904.1048v1 [math.CA] (7 Apr 2009).

17. A. Laforgia and P. Natalini, “On the asymptotic expansion of a ratio of gamma functions,” *J. Math. Anal. Appl.* 389, 833–837 (2012).

18. C. Mortici, “New approximation formulas for evaluating the ratio of gamma functions,” *Math. and Comp. Model*. 52, no. 1-2, 425–433 (2010).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30 January 2013	Technical Report	

4. TITLE AND SUBTITLE

Numerical Estimation of Information Theoretic Measures for Large Data Sets

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Michael B. Hurley and Edward K. Kao

5d. PROJECT NUMBER
1663

5e. TASK NUMBER
22

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-1169

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Air Force
AFLCMC/PZE
20 Schilling Circle, Bldg. 1305
Hanscom AFB, MA 01731

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
ESC-EN-HA-TR-2012-114

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A problem that has plagued the tracking community for decades has been the lack of a single metric to assess the overall performance of tracking systems. The authors' prior research has identified total conditional entropy as a useful measure to assess the overall performance of multi-target trackers and classifiers. The measure can be used to evaluate any system that can be formulated as an assignment algorithm that maps N classes of objects to M labels. The assignments from the decision system are compared to a truth data set to generate the total conditional entropy. This report focuses on work to generate error estimates so that the statistical significance of test results can be determined. Derivations of Wolpert and Wolf provide exact equations for calculating the first- and second-order statistics for the three fundamental entropy measures from sample data. The authors have restructured Wolpert and Wolf's equations into computer code that produces stable numerical results up to sample sizes in the billions. This code is provided as a MATLAB software package available from MIT Lincoln Laboratory.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	Same as report	84	